ARRL AMATEUR RADIO

COMPUTER NETWORKING CONFERENCES 1-4

PIONEER PAPERS ON PACKET RADIO 1981–1985

ARRL Amateur Radio Computer Networking Conferences 1-4

Pioneer Papers on Packet Radio 1981-1984

The American Radio Relay League
Newington, CT 06111

Printed in USA

$18.00 in USA
$20 in Canada and elsewhere
ISBN: 0-87259-022-4

First Edition

FOREWORD

This is a collection of technical papers presented at the first four ARRL Amateur Radio Computer Networking Conventions, They span the period 1981 through 1985, which were the formative years of amateur packet-radio development. These papers include a wide range of subjects. Some are theoretical; others cover practical applications. There is extensive treatment of protocols, software and hardware subjects.

The year 1984 sawthecompletion of the AX.25 Amateur Packet-Radio Link-Layer Protocol by the ARRL Ad Hoc Committee on Amateur Radio Digital Communication in September, and approval of the protocol by the ARRL Board of Directors in October.

All papers contained in this publication are unedited and are solely the responsibility of the authors.

David Sumner, K1ZZ
Executive Vice President

Newington, Connecticut
September 1985

Trademarks and service marks appearing in these papers are:

AMSAT is a registered trademark of The Radio Amateur Satellite Corporation.
AX.25 is a pending service mark of the ARRL.
CP/M is a registered trademark of Digital Research Inc.
Exar is a trademark of Exar Integrated Systems,, Inc.
Gunnplexler is a trademark of M/A-COM, Inc.
Heathkit is a registered trademark of the Heath Company.
ICOM is a trademark of ICOM Incorporated.
NOVRAM is a trademark of Xicor.
TRS-80 is a trademark of the Tandy Corporation.
TYMNET is a trademark of Tymshare.
UNIXisa trademark of American Telephone and Telegraph.
VIC 20 is a trademark of Commodore Business Machines, Inc.
Xerox is a trademark of Xerox Corporation.
Z80 and Z800 are registered trademarks of Zilog, Inc.

CONTENTS PAGE

First ARRL Amateur Radio Computer Networking Conference

October **16-17, 1981**
National Bureau of Standards
Gaithersburg, Maryland

Coordinator:
Paul L. Rinaldo, W4RI

Hosted by:

Radio Amateur Satellite Corporation & Amateur Radio Research and Development Corp.

AMATEUR PACKET NETWORK AGENDA

Paul L. Rinaldo, W4RI
1524 Springvale Avenue
McLean, Virginia 22101

Introduction

The American Radio Relay League (ARRL) is sponsoring this first international Amateur Radio Computer Networking Conference for several reasons. One is to recognize the innovative work that Canadian and U.S. amateurs have already done in packet radio. Another is *to* explore the possibilities of an integrated amateur packet network. Assuming that there is a consensus that a network can be developed, the third is to try to set up a framework for orderly growth.

This paper outlines my current thinking on some aspects of amateur packet radio networking. I have included a number of things that I believe should be considered at this conference and in the few months ahead.

Organization

In the past year, there has developed an informal group of packet radio leaders from different clubs. Fortunately, these individuals have been both advocates and doers. They have been in frequent touch with each other using various means of communication such as amateur radio, personal meetings, the mails and other methods. I'm speaking mainly of:

Stu Beal, VE3MWM	Hamilton, ON
Dave Borden, K8MMO	Sterling, VA
Larry Kayser; VE3QB	Ottawa, ON
Doug Lockhart, VE7APU	Vancouver, BC
Hank Magnuski, KA6M	San Francisco, CA

I would also like to be counted in this group. I'm sorry if I left anyone out who feels that he/she should have been included. The above list is meant to include the persons acting as **gateways** between their clubs and most of the others.

So far, this informal group has managed get the word around on new developments and has been able to move things along.

In any new enterprise, there is a tendency for someone to propose that the informal organization be replaced by a formal one. It is my feeling that this is a highly experimental and dynamic endeavor and that we shouldn't fix what's working.

Therefore, I suggest that we give at least tacit approval to this informal group and give them the ball on deciding when this leadership function needs to be formalized.

Our organizational energies should be focused, for the moment, on supporting the existing organizations. As an individual, you should see to it that your club attracts new members and helps them get on the air with packets. Virtually a.11 the clubs are grossly underfunded and could use help raising money, The club newsletters need more capable writers on a variety of packet topics,, In addition, writers need'to send top quality packet tutorials and technical articles to the ARRL publications *QST* and *QEX*.

Network Management Issues

Network architecture (structure, hierarchy, protocol, routing strategies) needs to be ironed out soon. The Network Architecture Seminar of this conference should help to move this subject along toward some type of agreement. I personally favor a two-tier system. At the lower level, there would be Local Area Networks (each with their own repeater), designed to fit the needs of the area. At the higher level, we would have a larger network (sometimes called an *internet*) which uses commonly agreed rules and can pass traffic via hf, satellite and terrestrial circuits. Also, I feel it essential to the network's growth that a new station (terminal or network node) be able to fire up without begging someone's permission.

Financing the network needs some fresh thinking. Undoubtedly, the local area networks will be financed locally as are the numerous Z-meter fm repeaters. However, the internet needs both local an.d network-wide financing. The latter category may include a number of the following possibilities:

- A network membership (users and supporters) fee on the order of $25 or more a year.
- A fund such as the ARRL Foundation to appeal to all radio and computer amateurs.
- Proposals for **seed** *money* to government and/or industry. There is a. joint AMRAD/ARRL proposal calling for funding of 8 hf packet stations in its early stages of consideration by the U.S. Government.

Applications of the network need to be thrashed out, The Applications Seminar of this conference should stimulate some new thought. For one thing, I think that the network should develop a capability of handling third-party traffic. Serious consideration should be given *to* handling teletypewriter (TTY) traffic for the deaf. Barry Strassler will present a paper on this subject. I believe that handling of deaf TTY traffic is feasible via a special type of computerized bulletin board (CBBS) developed by AMRAD called HEX (Handicapped Educational Exchange) which speaks both

ASCII and Baudot/Murray codes. In fact, we need to talk about the possibilility of interconnecting with *all* CBBS's in North America. This carries with it the problems of getting the individualistic system operators to understand the need to comply with the internet standards. Perhaps the knottier problem is how to screen all traffic to eliminate commercial and other no-no traffic before it is transmitted over Amateur Radio.

Acronymics

Acronymics is a made-up word meaning playing around with acronyms. We need some agreed names to call things within the network. I was recently advised to leave this subject to discussion over beer and pizza. Ignoring that, here goes.

First, I propose that the overall network be called AMNET. The term should specifically apply to the internet and generally to the various Local Area Networks (LANs) connected to it. AMNET is meant to be an umbrella term.

The internet will be made up of three transmission (sometimes called transport) systems, as outlined below:

- The satellite net, fortunately, has already been given an acronym: AMICON, standing for AMSAT International COmputer Network. In the interest of symmetry, maybe we should make the other transmission system acronyms end in CON.

- The net of vhf/uhf packet repeaters along the countryside or terrain could be called TERRACON.

- The high-frequency net uses the ionsphere to skip long distances, so why not dub this SKIPCON?

It would seem natural to follow commercial practice and call the stations that change from one net or medium to another *gateways.* I'm proposing to call these gateways: SKYGATE for satellite gateways, TERRAGATE for vhf/uhf gateways, and SKIPGATE for hf gateways. The need for SKYGATEs and SKIPGATEs is fairly obvious. But there could be TERRAGATEs too where there is need to change frequencies, data rates and perhaps protocols.

I suggest that we leave the naming of the LANs to the local sponsors. It may also be necessary to name some *virtual networks* that people cook up within the network. Eventually, I'd like to see all these names (frequencies and other parameters) published in a directory by the ARRL. The time will come quite soon when we will need a packet radio directory which lists all network facilities as well as individual packet radio stations. To kick this off, AMRAD is offering to collect the information and organize it in a directory form. We already have a CBBS Directory which could be included.

Internet Standards

If the internet is to work it must have agreed standards. There must be agreement on a wide set of particulars including protocols, routing strategies, radio frequencies, etc. Yet we need to temper this with maximum flexibility. There are a number of fundamental issues to be addressed. For example, do we want to look for government seed money and configure the network so that it can handle government traffic in emergencies; e.g., use ARPA's Internet protocol? Perhaps the other issues can be better handled according to the medium used:

Vhf/uhf Terrestrial Net Standards

This net should consist of a number of single-frequency vhf/uhf repeaters, each with a capability of working neighboring repeaters on the same frequency. This can go on for an unlimited number of hops, or the chain can be broken by changing to another frequency. Propagationally, the 144, 220 or 420 MHz amateur bands would be acceptable. I am inclined to push the 220-MHz band because it is underutilized. Influencing the choice of frequencies may be the signaling speed permitted by the Federal Communications Commission in the U.S. Current Rules permit only 1200 baud at 144 and 220 MHz and 19.6 kilobaud at 420 MHz.

What speeds should we use? In packet communications it is necessary to have a high enough signaling speed to handle the volume of traffic without the network going critical and being tied up permanently with retries. That says, "the higher speed the better." But, raising speed means increasing bandwidth (all other things equal) which means higher transmitter power for an acceptable signal-to-noise ratio. Certainly, 1200 baud is not sufficient for such a net. 2400 and 4800 baud are not that much better. 9600 becomes marginally usable. From here, the American National Standards Institute (ANSI X3.36-1975) pegs the standard speeds at integral multiples of 8000 bits per second, so the next speeds are: 16, 24, 32, 40, 48, 56 kilobits, etc. ANSI shows 16 and 56 as "selected standard signaling rates" and recognizes 48 kilobits per second as a recognized standard for international transmission. Thus, it looks like the choice for the high end is either 48 or 56 kb/s. The low-end choice seems to be 16 kb/s. I was pushing 48 kb/s because of its international blessing by the CCITT, but I have learned that very few circuits have been implemented at that speed and that modems are extremely rare/expensive. I suggest that we look very closely at 56 kb/s as a proposed standard. That still gives us a modem problem to be solved by amateur ingenuity.

If we want to operate such speeds at 220 MHz, we will need to petition the FCC for a Rules change. Probably the best approach is to request a Special Temporary Authority (STA) as a first step. I believe that we want to ask for permission to use

up to 56 kb/s in a portion of the 220-MHz band. You may have noted that the Canadians already have Department of Communications permission to use various bandwidths up to 100 kHz in the 220-MHz band. Unfortunately, there is no simple method of equating bandwidth to signaling speed because things change with different modulation schemes. There has been a tendency for hams to think of fsk for transmission of data because we are used to sending RTTY that way. The problem is that fsk gobbles up too much bandwidth. Phase-shift keying (psk) conserves bandwidth, particularly if we can use quadraphase psk. There is much work to do on designing practical modems for these data rates. Commercial (say 56 kb/s) modems are far outside our price range. I'd like to see someone look into the design feasibility of a qpsk modem operating at a 10.7-MHz carrier frequency for the internet repeaters.

As for where in the 220-MHz band we should put these repeaters, I circulated an informal letter on this subject in May of this year. If you wish to research this problem, you should look at the Canadian Amateur Bands,[1,2] U.S. Amateur Radio Frequency Allocations,[3] and the ARRL Vhf-Uhf Advisory Committee 220-225-MHz band plan.[4] Study of these references reveals that the 220.0-220.5 subband is not usable because FCC rules do not permit repeaters. The FCC permits repeaters above 220.5 MHz. Starting at 221.9 MHz there is a weak-signal guard band, EME (moonbounce), propagational beacons, weak-signal cw, calling frequency, general operations (cw/ssb), as well as fm repeaters and simplex channels to use up the higher part of the band, as outlined in the band plan. So, that narrows packet repeater operation down to the subband 220.5 to.221.9 MHz in the U.S. This applies only to the internet repeaters or possibly local area net repeaters which use data rates above 1200 baud (with FCC STA or rules change, of course). Local repeaters using 1200 baud could operate on fm voice repeater or simplex frequencies in the 144, 220 or 440-MHz bands.

The proposed 220. 5-221.9-MHz internet packet repeater subband needs to be broken down into channels. Here's my first cut; I'd welcome some comments:

a. Channelize on 100-kHz intervals, e.g., **220.6, 220.7,** etc. This will allow us to run up to 100 kHz bandwidth per channel. It will probably be necessary to keep two channels operating in the same area at least 400 kHz apart. Packet receivers may use fm broadcast 10.7-MHz i-f transformers because of their low cost. Although fm broadcast band channel spacing is every 200 kHz, it is not usual to have adjacent channels allocated in the same area.

b. As the arguments over whether to use simplex or duplex packet repeaters are not over, it might be prudent to look at the possibility of designating some duplex channels within the 220. 5-221. 9-MHz subband,, If the duplexer possibilities permit this, it would seem logical to have several duplex channels with inputs at one end and outputs at the other end of the subband. Then, the simplex packet channels could go in the middle:

Frequencies	Proposed Use
220.6/221.6	Packet repeater pair
220.7/221.7	"
220.8/221.8	"
220.9	Packet repeater simplex
221.0	"
221.1	"
221.2	"
221.3	"
221.4	"
221.5	"

There is an immediate need for a group of people to work o:n terrestrial vhf/uhf repeater standards and hardware design. Because of the scarcity of commercial 220-MHz equipment with up to 100 kHz bandwidths and quick turn-around time, it appears that most 220-MHz packet repeaters will be homebrewed. Doug Lockhart, VE7APU has circulated an informal paper which proposes certain design criteria. His basic idea is to come up with a single-board repeater that can be easily replicated across the continent. We need someone to undertake this design effort on a priority basis.

AMICON Standards

The focus of this work has been on the use of the data communications special service channel (L2) on the AMSAT Phase III satellite. There is already a committee in place headed by Hank Magnuski, KA6M. They have already drafted several generations of AMICON specifications which have been circulated to those directly involved.

Hf Standards

My concept is to have at least 8 hf gateways in the U.S., and about 6 in Canada. Within the bounds of ionospheric propagation, all should be able to talk directly to each other. If not, some can relay.

I would like to see the hf net run automatically, without human operators involved in sending and receiving packets. Computer control and digitally controllable rigs such as the ICOM IC-701/720 and the Collins KWM-380 permit this. A signal plan in software would provide the time slots with frequencies, antenna orientations and other information required for communications between the then-active stations.

Now, on to hf signaling speeds. Here is an area where we will. have to do some experimentation. The enemy is *multipath.* If you could always operate at the maximum usable frequency (muf) for the path, you could send at almost any speed because there

would be only one path through the ionospheric layer. On the ham bands, it is not at all certain that you will be operating at or near the muf. Presently, the hf bands are allocated every octave, the exception being the 21-MHz band which falls between the 14- and 28-MHz bands. The addition of the new WARC bands at 10 and 18 MHz will help this situation for the longer paths. There remains an octave gap between the 3.5- and 7-MHz bands which will make it difficult for the shorter hf paths where multipath is a limiting factor. (It would be nice to have a frequency near 5 MHz that could be used by hams for packet operations.)

The worst-case situation on hf for multipath is in the vicinity of **100** km (near-vertical-incidence paths). Here, the maximum expected delay difference at frequencies below the muf could be about 8.5 ms, according to Salaman work on Multipath Reduction Factor (MRF).[5,6]

MRF or delay difference is manifested by a new bit and old bit overlapping each other. The question becomes one of how much overlap can be tolerated before the signal is no longer readable. I suppose that the answer has a good deal to do with the decision-making process used in the demodulator and signal-processing circuits. Common sense would seem to lead one to conclude that 50% overlap is about as much as a receiving system could put up with. Laport says that the signal is mutilated when the prominent delayed wave in the multipath group surpasses 20%.[7] I suspect that a better number is somewhere between these values.

Fig. 1 shows the maximum signaling rates in bauds as a function of path distance and operating frequency with respect to the muf. It is derived from Salaman's work. The conversion used to change delay in ms to bauds is based on a 50% overlap of code elements. If you wish to believe the more pessimistic 20%, then you can divide these rates by 2.5.

I suggest that we recognize 75, 150 and 300 baud rates as standard for hf packet operations. Also, we should begin experiments with 600 baud under an STA from the FCC in order to determine whether it should also become standard under a rules change. Initially, hf packet testing should use 75 baud to iron out the other bugs. Then, testing of higher speeds should proceed under valid test methods capable of sorting out multipath from inadequacies of demodulators, etc. Operationally, it makes sense to operate at the highest speed at the time, consistent with good copy. Assuming that we find the speed will vary under different path conditions, we should consider making the speed adaptive. By adaptive, I mean that signaling speed should be under software control so that the computers at each end can decide on which data rate is best for the prevailing conditions.

Multipath distortion is not the only problem facing us on hf. Other types of fading, other signals, man-made noise and natural noise all conspire to prevent us from sending perfect packets through the ionosphere. Hams will make a substantial

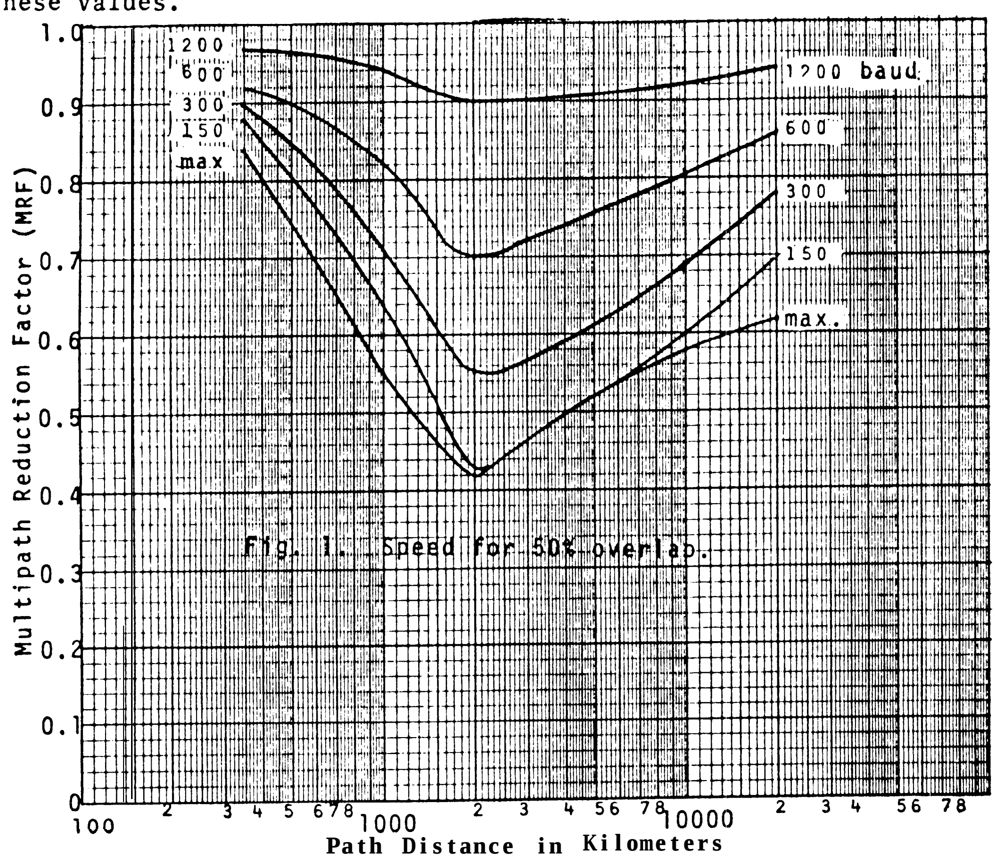


Fig. 1. Speed for 50% overlap.

contribution to technology if we can make this work reliably. Simply sending packets over an hf circuit will result in many retries (when conditions are not perfect). Nonetheless, some of that should be done in order to get a feel for the problem. However, it now looks very likely that we should use some type of forward error correction (FEC). An FEC system called AMTOR, based on CCIR Recommendation 476, is currently being tested overseas.[8] A small number of U.S. amateurs were granted an STA to experiment with AMTOR in November 1980 by the FCC.[9] Jerry Dijak, W9JD is currently developing an hf FEC system.[10,11] There remains much work to do, both theoretically and experimentally. There are some interesting questions. One is, why do we keep throwing out packets which have one or more errors in them? Is it practical to compare two or three pretty good copies of a packet in memory and make a perfect one?

We need some standard operating frequencies for hf packet communications. At this time, I believe that we need two frequencies per hf band -- one for network operations, one for direct use between individual stations (for experimentation, traffic handling, etc.).

In February, 1981, I mailed an hf packet frequency survey request to a number of individuals in locations covering North America. Unfortunately, the response was not what I had hoped. Rut, I wish to thank Bill, W4MIB and Pete, N5TP for the considerable number of hours that they monitored looking for activity on the RTTY bands. Their results, combined with mine, showed a fairly consistent bell-shaped curve within the RTTY segments. This led to the general conclusion that there were some good spots just inside the low and high ends of each RTTY subband. In addition to monitoring for on-the-air activity, research was done on published or other known usage.[12] Also taken into account were: U.S. Fl allocations,[3] Canadian Fl allocations,[2] the "Considerate Operator% Frequency Guide,"[13] and the IARU Region 1 HF Band Plan.[14]

<u>80 Meters</u>:

3500- 3775	U.S. Fl allocation
3500- 3725	Canadian Fl allocation
3610- 3630	Considerate Opr's Freq. Guide
3580- 3620	IARU Region 1 HF Band Plan Fl

Published or other known usage:

3580	W1AW cw bulletins & code practice
3583	Southwest CW Traffic Net
3585	Missouri CW Net
3587.5	Louisiana RTTY Net
3590	Empire Slow Speed Net
	Third Region Net
	Washington Section Net
3595	Georgia State Net
3596	Pine Tree Net
3598	Southern California Net
3600	Kentucky CW Traffic Net
	Kentucky Slow CW Traffic Net
3602	First Region Net
3605	Buckey Net RTTY
3610	Eastern Pennsylvania CW Net
	Kansas CW Section Net
	Pennsylvania Training and Tfc Net
3615	Louisiana Amateur Net
3617.5	Virginia Specialized Comm. Net
3620	AMSAT - RTTY Net
	Georgia Emergency RTTY Net
3625	W1AW RTTY Bulletins
3630	Kentucky RTTY Net
	Northern California Net
3633	New Hampshire Net
3635	Idaho Montana Net
	Tennessee CW Net
3637.5	RTTY Autostart

<u>40 Meters</u>:

7000- 7150	U.S. Fl allocation
7000- 7150	Canadian Fl allocation
7090- 7100	Considerate Opr's Freq. Guide
7035- 7045	IARU Region 1 HF Band Plan Fl

Published or other known usage:

7040	Eastern Canada Net
	Hit & Bounce Net
7045	Ontario Southern Net
	IARU Region 1 HF Band Plan for SSTV
7090	Forty Meter Interstate RTTY Net
	Gater Net
7095	W1AW RTTY Bulletins

<u>20 Meters</u>:

14000-14200	U S Fl allocation
14000-14100	Canadian Fl allocation
14080-14100	Considerate Opr's Freq. Guide
14080-14100	IARU Region 1 HF Band Plan Fl

Published or other known usage:

14075	RTTY autostart
14076.5	Canadian packet beacons
14080	W1AW bulletins & code practice
14082.5	Heath computers, autostart
14095	W1AW RTTY bulletins

<u>15 Meters</u>:

21000-21250	U.S. Fl allocation
21000-21100	Canadian Fl allocation
21090-21100	Considerate Opr's Freq. Guide
21080-21120	IARU Region 1 HF Band Plan Fl

Published or other known usage:

21095	W1AW RTTY bulletins

<u>10 Meters</u>:

28000-28500	U S Fl allocation
28000-28100	Canadian Fl allocation
28090-28100	Considerate Opr's Freq. Guide
28050-28150	IARU Region 1 HF Band Plan Fl

Published or other known usage:

28080	W1AW bulletins & code practice
28095	W1AW RTTY bulletins

After review of all information available, my recommendations for specific hf packet frequencies are:

Band	Direct	Network
80 meters	3612.5 kHz	3627.5 kHz
40 meters	7092.0 kHz	7098.0 kHz
40 meters R1*	7036.0 kHz	7044.0 kHz
20 meters	14076.5 kHz	14098.0 kHz
15 meters	21092.0 kHz	21098.0 kHz
10 meters	28092.0 kHz	28098.0 kHz

**R1 frequencies are for use in ITU Region I and for Transatlantic packet communications.*

Over the past few years, there has been a virtually complete phase-out of 850-Hz shift for hf fsk. The hf fsk standard is now 170 Hz. This has been somewhat of a mixed blessing. On the positive side, 170-Hz shift occupies less bandwidth thus conserving spectrum in that sense. However, at 850-Hz shift, there was considerable decorrelation of the mark and space frequencies. In other words, the mark and space frequencies were so far apart that they tended to fade independently. This opened up the possibility of processing these two signals as two separate diversity branches. Diversity combining could give equivalent gains of something on the order of 8 dB or so depending on a number of factors. In a way, this was moot because amateur RTTY demodulators did not take advantages of this decorrelation when 850-Hz shift was used. I bring this up because we have another shot at it. Clearly, 170-Hz shift is not what we need to run data rates such as 300 and (possibly) 600 baud. Bob Watson, a design engineer who is working with state-of-the-art demodulation techniques, has given this problem a great deal of thought. He is proposing that we go to 600-Hz shift. This does a number of good things. 600 Hz is wide enough that mark and space tend to be decorrelated most of the time. It permits keying speeds up to 600 baud. It would allow us to use synchronous transmission and reception. Frequency diversity and synchronous detection can provide considerable gain.

Local Area Net Standards

For the most part, LAN standards seem to be developing along the lines of 1200 baud, Bell 202 modems, 2-meter simplex repeaters, using the VADCG terminal node controller board for the individual station, etc. Much of this has to do with the availability of the VADCG TNC boards and quantities of Bell 202 modems. The 1200-baud data rate is also the highest speed presently authorized by the FCC for the 144 and 220 MHz bands.

Things don't necessarily have to stay in this same pattern. In fact, there will develop a number of reasons why we should try some different techniques. If speed is a problem at 144 and 220 MHz, of course we can move to 420 MHz where the FCC permits 19600 baud. Someone can make a pitch to the FCC to change their rules to allow higher data rates at 144 MHz and above. In fact, the ARRL has already done so under petition #3788. The availability of higher-speed surplus modems or an easily reproducible printed-circuit board designed by amateurs could make the higher-speed modem picture a bit brighter. A new board to replace the VADCG TNC could change things, possibly by reducing the cost. In other words, there is more than enough room to experiment.

My basic point is that we need some commonality to help local networks come about but need to encourage local experimentation and innovation.

Conclusion

I hope that the foregoing information and recommendations contribute to the thinking processes in the development of amateur packet radio networking. Please understand that nothing that I have presented is cast in concrete. We are all learning.

The time to design a packet network is now. Many of the people who will do the work are at this conference. We can muster the talent and resources needed to do the job.

I would like to thank the ARRL, National Bureau of Standards, AMRAD membership, AMSAT membership, the Bureau Radio Amateur Signal Society and numerous individuals who have made this conference possible.

References

[1]Hesler, "Canadian Newsfronts: DOC Creates New Amateur License Class," *QST*, December 1978, p.61.
[2]ARRL, "1981 The Radio Amateur's Handbook," p. 1-4.
[3]*ibid*, p. 1-5.
[4]ARRL, "The ARRL Repeater Directory," 1981-1982 ed., p.18.
[5]Salaman, "A New Ionospheric Multipath Reduction Factor (MRF)," *IRE* Trans. on *Com. Sys.*, June 1962.
[6]ITT, " Reference Data for Radio Engineers," 6th ed., 1975, p.28-9.
[7]Laport, "Radio Antenna Engineering," 1967, p.212.
[8]Martinez, "Amtor, an Improved Error-Free RTTY System," *QST*, June 1981, p.25.
[9]FCC Special Temporary Authority to K4PA, K3FLS, W3KET and KB6BT, *AMRAD Newsletter*, January 1981.
[10]Dijak, "Data Communications with Forward Error Correction," *AMRAD Newsletter*, August 1981.
[11]Dijak, "The W9JD FEC System," *AMRAD Newsletter*, October 1981.
[12]ARRL, "Net Directory," 1980-81 ed.
[13]ARRL, *QST*, January 1981, p.47.
[14]Eckersley, "Amateur Radio Operating Manual," RSGB, 1979., p.53.

SOFTNET - PACKET RADIO IN SWEDEN

Jens Zander, SM5HEV
Department of Electrical Engineering
Linkoping University
S-581 83 Linkoping, Sweden

1.7

Abstract

An experimental packet radio network is under construction at the University of Linkoping, Sweden. The network is distributed and all nodes are programmable via the network during normal operation. This concept gives full flexibility at all levels. Experiments at low levels, such as access schemes, as well as at high levels, such as routing and flow-control, are possible. Finally, the implementation of the network is sketched.

I. Introduction

In Sweden, just as in the US the personal computer market literally exploded in the late 70's. The market was, and still is, dominated by computers of Swedish origin, Luxor ABC80, as well as the american brands, PET, Apple and Radio Shack. At an early stage, the computer clubs at the major Universities took a leading part in the evolution of personal computers. In 1975, the LYSATOR club at the University of Linkoping had already designed and was distributing a minicomputer kit using the IMP-16 chips for personal use. Also, many radio amateurs got involved, and their need for computer communication made the Swedish Telecommunication Administration finally allow the use of ASCII for transmissions in the amateur bands.

The idea to organize these computer communication attempts substantiated as one of us attended a course Computer Networks at USC in Los Angeles. Back in Sweden? the first sketches of the proposed wide-band packet radio system were quickly adopted by radio amateurs, the LYSATOR club and other persons at the University of Linkoping. Soon a close cooperation with a research project dealing with digital mobile radio communication was established. This project was among other related issues concerned with mobile packet communication, primarily access schemes.

Since then, half a year has passed. Students, both graduate and under-graduate, have become involved in the Softnet project. The aim of this project is to construct a distributed packet radio network, operated by radio- and computer-hobbyists, for experiments with routing, flow control, DDP and many other things. The swedish Telecommunication Administration has shown interest in the project, and indicated that they will grant permission to operate such a network. The frequency band used will be 432 MHz,

II. Softnet concepts

During the last years several amateur packet radio networks have evolved, mainly in Canada and the US. These networks consist of a number of users nodes and a repeater. All users communicate via the repeater and are not allowed to communicate directly with each other. This is an example of a centralized network (Figure 1a). The commercial telephone networks are other examples of centralized networks. The advantage of this kind of network is its simplicity of operation. Routing is trivial, no one has to know exactly where the receiver is located. Packets are just forwarded to the repeater. There are, however, several disadvantages with a system like this for amateur experiments. A network of this kind is almost condemned to be a local one, unless provision for inter-repeater communication is made. When the repeater fails, no traffic exchange is possible. This places a heavy burden on the person responsible for repeater operation.

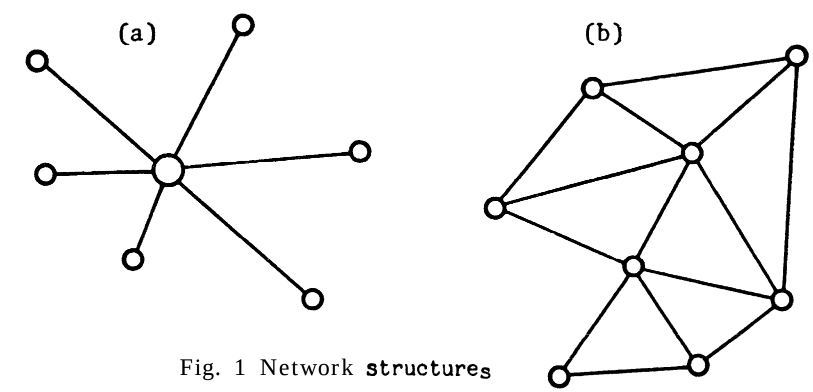


Fig. 1 Network structures

An alternate, more flexible, solution is to use a distributed network (Figure 1b). Here every node is able to communicate with all other nodes within range. Also, every node acts as a repeater for packets, forwarding them in the 'right' direction towards its final destination. Examples of networks of this kind are the PR-net and the ARPA net. Such a network eliminates the responsibility problem and makes the system more robust. If any node fails, there may be another path for packets to take. It is in the owners own interest to keep his node running. A node can be installed almost everywhere. The only requirement is that another network node is within radio range.

When constructing an experimental network without really knowing what future demands are going to be, it is essential that the network is 'soft' at all levels. This is achieved by making the nodes programmable. Full flexibility is achieved by allowing nodes to be programmed via the network during normal operation. This is the Softnet concept. Thanks to this flexibility one part of the network can be, for instance, operated as a centralized network one day, and as a distributed network another day. Node programs are forwarded in packets, just as ordinary data.As will be seen, this makes it natural not to distinguish between programs and data, a feature which is common in many modern programming languages.

III. Problem identification

There are several problem3 that arise in a distributed network. Consider the network in Figure 2. The lines between nodes represent possible two-way radio paths. A trivial task for a network is to forward a packet from, say node A to node F. We can immediately see that there is no direct two-way path between those nodes. This means that the packet has to reach its destination in a multi-hop fashion. The question then arises, which of the nodes to use as intermediate repeaters. We have a routing problem. This is a very interesting problem in radio networks since we can change the topology of the network by simply increasing or decreasing the transmitter power. Furthermore, node B may be heavily loaded by other traffic which makes node D more suited as the first repeating node. Let us for a moment assume that node B is chosen by node A to be the first repeating node. We are using a single, wideband, radio channel which is shared by all nodes within range. The nodes have to agree upon some scheme or algorithm by which this sharing is made possible. We have here a local communication problem or a channel access problem. Finally when a new node, G, is installed it should quickly become a full member of the network. This is a problem of network control.

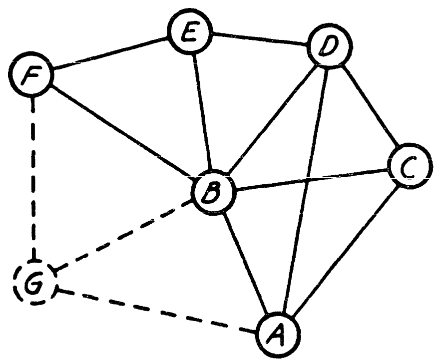


Fig.2 Network example

We have by this little example identified three major problem areas, Routing, Access, and Network control. Being strict, to achieve optimum performance these cannot completely be held apart [3]. It is, however, reasonable to believe that, if these problems are treated separately, near optimum results will still be achieved.

IV. Network programming

Traditionally, computer networks utilize layered structures of protocol3 , i.e. procedures by which messages are handled at different levels. Typically, a protocol uses a number of parameter3 which can be set to make the involved stations react in a specific way. This method is well suited for fixed networks and will easily handle all the situations for which it is designed. In an experimental network, surely a lot of unpredictable situations will arise and this demands another kind of solution.

An obvious remedy is simply not to specify any specific procedures or protocol3 at all. Instead, we define a language in which possible solutions can be expressed, This language will define a logical (node-) machine that can be instructed to perform function3 defined by the (remote) user. In Softnet, FORTH was chosen as the control language. This language has several properties that makes it highly suitable as network language in an experimental environment. FORTH is interpreted and its source code is extremely compact . The greatest advantage is, however, that FORTH is incremental in that new language constructs can be defined by the users. These new constructs are used in as general a way as the basic primitives.

In Softnet each node acts as an interpreter of packets containing FORTH statements which are immediately executed. The statements are typically 'treat the rest of the packet as data and forward it to node B', but they can also define new function3 as 'forward all my packet3 to node C'. A small example may clarify the general idea. If we ret-urn to the network in figure 2, we are now going to let the network perform a little more difficult task. Suppose we want to implement a point-to-multipoint connection, say between node A and both nodes C and D, using B as an intermediate repeater. A possible solution is to transmit the packet3 shown in figure 3 to node B. In the first packet we define a command (FORTH word) called SPLIT. SPLIT is defined to duplicate a packet (DUP) and forward one copy to C (C SEND) and one copy to D (D SEND). This definition is stored in B and is now ready to be used. The second packet makes use of the SPLIT word. The shaded area here represent3 a field treated as data, Finally, we can remove the definition to save some of the memory space in node B by transmitting the third packet in figure 3.

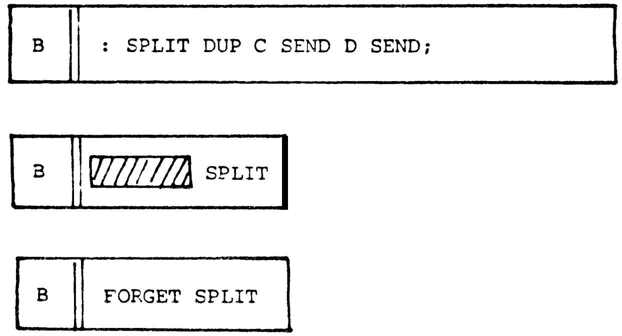


Fig. 3 Example of packets creating new network facilities

By this example we can see that Softnet not only can be programmed, but in fact programming is regulary used to perform all kinds of tasks. The network can be considered as a giant distributed computer. Unintended interaction between users is avoided by permitting them to store their functions temporarily in private memory areas in the distant nodes. The network normally contains primitives to implement basic facilities at boot-up time. Examples of such facilities are datagrams, virtual calls and file transfers. The owner is, however, able to create his own counterparts of these facilities or to extend them for his personal use.

V. Softnet implementation

Presently the first nodes for Softnet are under construction at the University of Linkoping. The node functions are implemented as three concurrent processes (Figure 4). Each of these processes consists of a complete Forth interpreter that interprets packets (NODE and LINK processes) or the terminal input stream (USER process). The LINK process handles local communication. It controls the packet transceiver and stores packets. Routing and network control are dealt with by the NODE process, whereas the USER process constitutes the user interface [2].

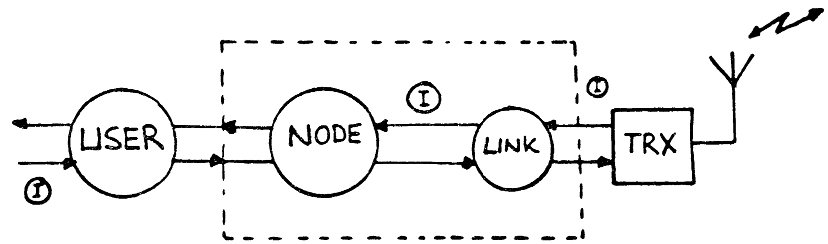


Fig 4 Node processes

These processes are implemented in a two-processor microcomputer system, utilizing the Motorola 6809 CPU. This CPU was chosen because it is virtually a Forth machine in silicon. Figure 5 shows a block diagram of the node. The NODE and USER processes reside in the main processor, whereas the LINK process is implemented in the Packet Radio Interface (PRI). The PRI contains a packet buffer shared with the main system. The packet radio transceiver interfaces to the PRI via a synchronous communication chip (SSDA) and a parallel interface (PIA). The basic primitives (a collection of FORTH words) and the interpreter are contained in PROMs to enable node bootstrap. It is, however possible to distribute revised versions of the basic words via the network for RAM storage.

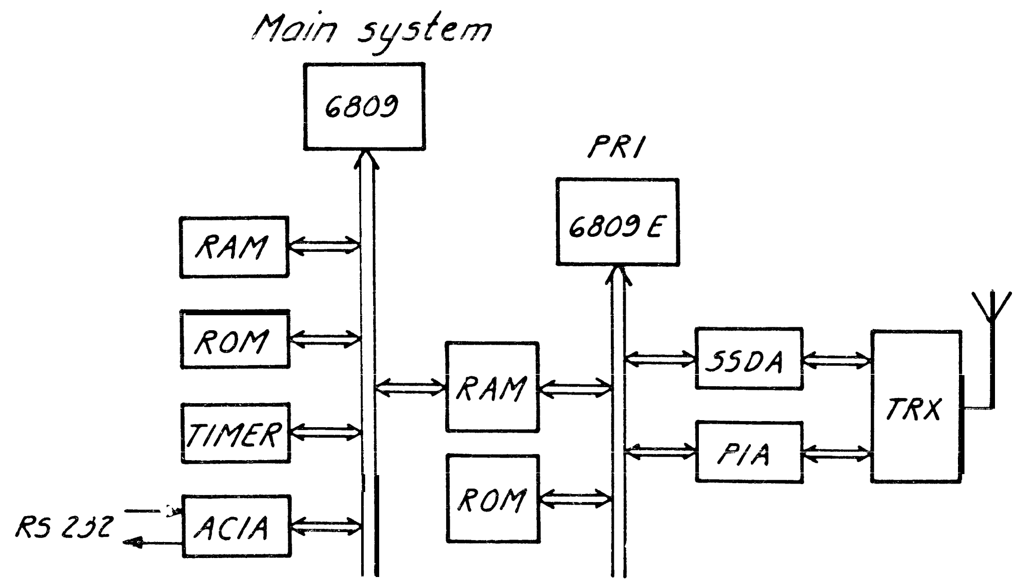


Fig. 5 Node - Block diagram

The packet radio transceiver is a mixture of standard and special purpose designed components. Figure 6 shows a block diagram of the transceiver. The Receiver section uses a standard BC IF strip and detector. The bit rate used in the system is 100 kbps with clock recovery coding for synchronous transmission. The signal strength reading (S) is digitized and is available to the PRI. The transmitter section consist of a PLL FSK modulator at 30 MHz. FSK modulation was chosen for its simplicity of design and high interference suppression. The conversion to the actual frequency band, 432 MHz, is performed by a standard transverter. The transmitter section of the transverter is modified to enable digital power control. The output of the transmitter is less then 10 W. The T/R switching is performed in a strip-line PIN diode switch (described in QST May 1981). Provision is also made for testing a carrier synchronous DPSK modulation system.

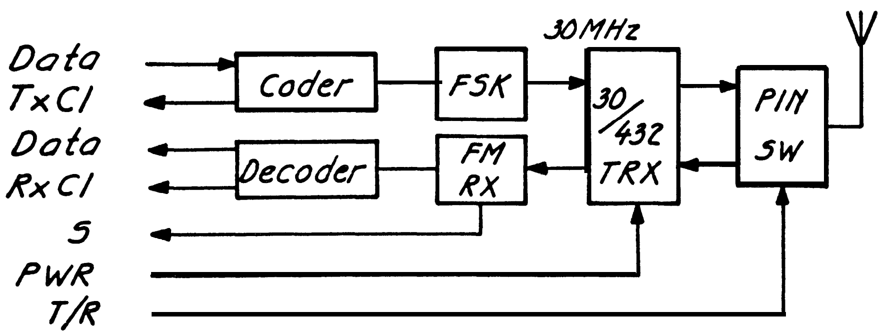


Fig.6 Packet Radio Transceiver - Block diagram

VI. Conclusions

The Softnet concept is well suited for amateur radio computer networks, It provides the flexibility that a network designed for experimental purpose3 should have. The programmability gives all users the opportunity to conduct their own experiment3 during normal network operation. The network is highly robust against failures. New users may easily enter and leave the network. Based on these properties, it is our belief that the Softnet concept is a viable approach to amateur computer communications.

References

[1] Persson,I., Forchheimer, R. : Design Considerations of a Distributed Packet Radio Network using the Amateur Radio bands, Internal Report, LiTH-ISY-I-0408, May 1980

[2] Zander,J., Forchheimer,R. : Preliminary Specification3 for a Distributed Packet Radio Network for Computer- and Radio-Amateurs, Internal Report, LiTH-ISY-I-0424, January 1980

[3] Silvester,J. : On the Spatial Capacity of Packet Radio Networks, Dissertation, UCLA-ENG-8021, May 1980

[4] Barber,D.W. et al. : Computer Network and their Protocols, Wiley 1980.

AN EXPANDABLE MICROWAVE NETWORK FOR MULTIMODE COMMUNICATIONS

Dave Ingram, K4TWJ
Eastwood Village #1201 South
Rt. II, Box 499
Birmingham, Alabama 35210

The projected network described in this paper was originally conceived with the purpose of interlinking communities and cities on a broadband basis. Numerous other capabilities, however, were soon included to permit almost direct compatibility with future communication expansions. The resultant network is highly flexible: it may be instigated between adjoining communities and/or cities, with additional networks being implemented in other areas and interlinked as desired. Communication modes which can be handled by the network are limited only by users desires and their respective modes. A basic outline of the microwave network shown in Figure I, and an overview of its operations follow.

Network Philosophy

The primary purpose of the microwave network is providing emergency communications between areas or cities normally separated by a distance greater than their normal 2 meter communications range. Secondary communications capabilities should be considered at installation time, however, since path losses and overall network bandwidth are directly related. The number of "dumb" or "passive" microwave repeaters will be determined by distance and terrain between associated cities, each accepting responsibility for their part of the link. Existing 2 meter repeater groups and councils can respectively provide finances and frequency/code coordinations. Two transceivers are shown connected to each microwave port: one preset on the "primary" frequency and the other scanning an approximate I mHz range of the 2 meter band (Exception : all secondary transceivers realize primary-frequency lockout). "Secondary" transceivers are under microprocessor control , permitting frequency scanning, spread spectrum operation, tone control of transceiver functions (enable/disable, lockout, connect to mailbox, etc.). The network could initially develop between any two mutually-agreeable areas (each preferably with at least two local 2 meter repeaters, since this would confine costs of microwave link additions). Additional areas could join the existing network by financing their respective part while exhibiting their benefits to existing network users. Assuming a similar network is also instigated in other and more systems may "grow" until an overall network merge is warranted and instigated. Additional networks may, likewise, grow and merge with the existing system as desired. Further expansions may include "spurs" and subnetworks as desired.

Satellite Interlinking

Continuing the network a step further, interlinking with the OSCAR Phase IV geostationary satellites could provide full hemisphere to complete world coverage for compatible mode users (projected date: 1986). The outline for this concept is shown in Figure 2. OSCAR Phase IV is slated to include several concepts applicable for data communications. Some of these features are dedicated channels , tone controlling and mailboxing. In some instances, a microwave network port may interface with an OSCAR earthbased transponder. Other times, a separate network-to-satellite earth based station will be required. The criteria will, naturally, be determined by geographic locations of microwave links.

OSCAR satellites necessarily utilize narrowband modes such as SSB or CW, however a microwave network should utilize a constant carrier mode such as FM. The key to compatibility between these modes is Constant Amplitude Single Sideband, or merely PLL-SSB. This concept, which was developed in Europe 4 or 5 years ago, employs a variable amplitude in the normally suppressed carrier. Carrier amplitude is miniscule during modulation, but increases to full power during breaks of speech (after passing through the microwave network, the carrier may be fully removed - resulting in conventional SSB). Finally, total microprocessor control is employed for the link; its preprogrammed functions being available for call-up by coded tones.

Technical Aspects

The concepts associated with microwave links are, in several respects, unlike those employed in conventional VHF repeater links. Bandwidths of microwave systems, for example, are typically .5 to 4 mHz. Output power levels are noticeably lower, with large parabolic dishes providing signal gain capabilities. Conventional superhetrodyne techniques are also altered: each microwave transmitter runs continuously,, with a small portion of its output power being directed to its receiver's "front end" to provide a local oscillator signal. Transmit frequencies of communicating units are then offset by a difference equal to the desired I F (center frequency). This arrangement may be visualized with the aid of Figure 3. All microwave units are originally transmitting on their hypothetical resting frequency. An incoming signal on 146.00 mHz shifts the transmitting unit 146 mHz (a second signal on 146.50 and a third signal on 146.80 would appear as subcarriers

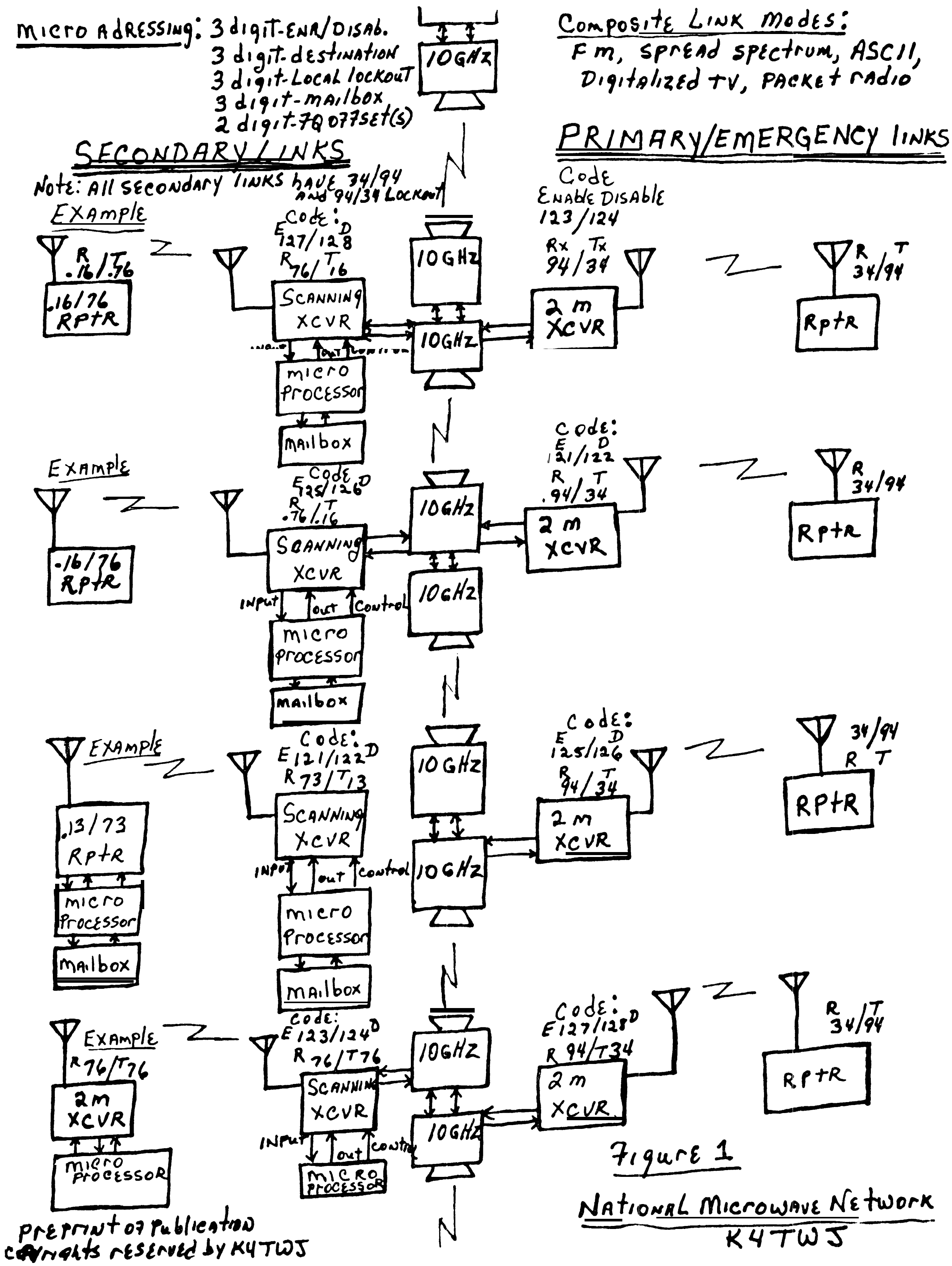


Figure 1

National Microwave Network

K4TWJ

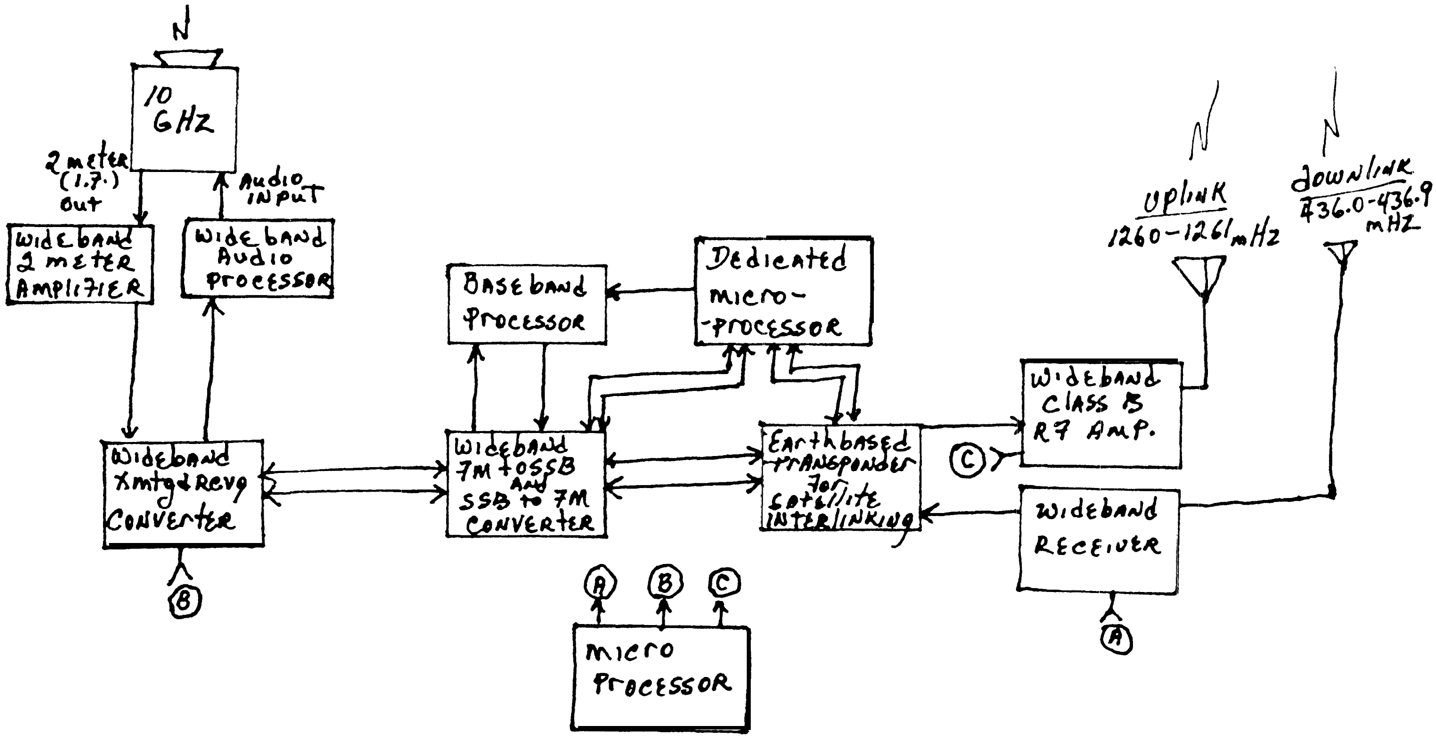


Figure 2 OSCAR Phase IV interLink for National Microwave Network. Compatability is accredited to PLL-SSB concepts.
master microprocessor monitors "control frequency" of 2 m. And activity on 2 m. bandpass to enable satellite link if: A prestablished count of signals on 1) downlink 2) uplink 3) 2 m. bandpass aren't exceeded. If justified, then proper "followup sequence" of 2 m. "control signals" on specific frequency will access satellite link.

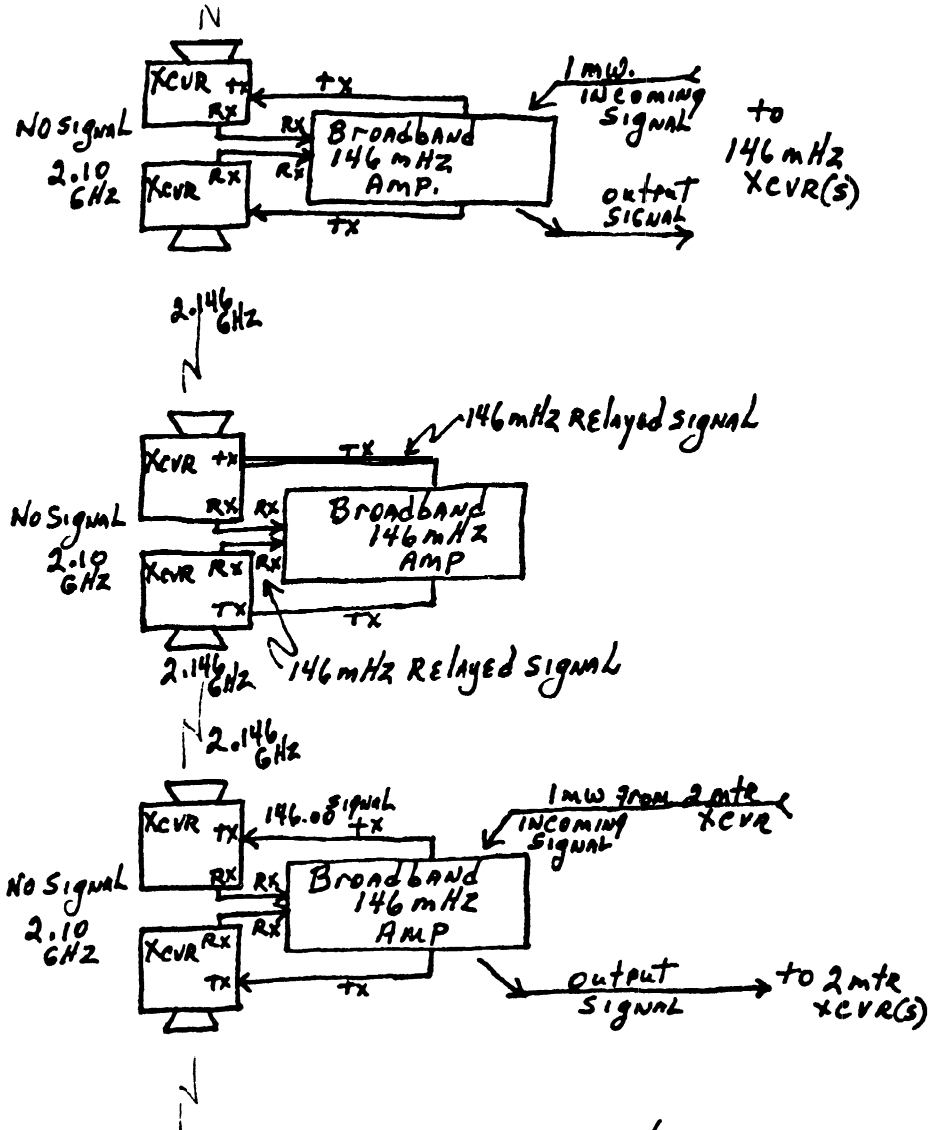
K4TWJ

of the original signal, until the 146.00 disappeared. The 146.80 signal would then be a subcarrier of the 146.50 signal would then be a subcarrier of the 146.50 signal). Assuming a "dumb" relay is required between ports, it would receive the 2.146 gHz signal, hetrodyne to .146 gHz, IF amplify the signal(s) and apply it to the associated transmitter. The 2.146 gHz signal would then be received at the subsequent microwave port, converted to 146 mHz and applied to a (broadband) amplifier. That (I. F.) amplifier's output would feed the "next" 2. I gHz transmitter and the 146 mHz transceiver (while also accepting 146-band input signals from the 146 mHz transceiver). The microwave% overall bandwidth could easily expand to I mHz, as necessary, with all data/tones being moved in a conventional manner. All operations and frequency determinations of network - located 2 meter transceivers are under microprocessor control. This means that port-available signals may be selected or rejected by tone control, as desired. Preprogramming of the microprocessor establishes basic network standards.

Two microwave! bands are prime candidates for network links: 2.1 gHz and IO gHz. Gunnplexers are readily available for 10 gHz systems, however their individual-link range is limited. Similar units for operation on 2.1 gHz will soon be available from Universal Communications, P. 0. Box 339, Arlington, Texas 76010. The 2.1 gHz units are 100 mw or I watt., as required. cost of IO gHz Gunnplexers are approximately 115 dollars each. Cost of 2.1 gHz units are approximately 170 dollars each.

Returning to Figure I and applying previously acquired knowledge, a brief technical discussion will now be presented.

Assume an amateur operating on 146.76 desires to contact a distant repeater on 146.76 mHz. A .76 signal with PL tone is used for connecting the scanning transceiver into the network. Notice the distant '94 transceiver employs "lockout", preventing accidental access. Another 3 digit code "brings up" the desired distant '76 scanning transceiver , with subsequent microprocessor control establishing operating parameters for accessing that area's '76 repeater.



K4TWJ

Figure 3 Operational Concepts of the National Microwave Network
middle unit is considered "dumb" or passive repeater, with intelligent ports being located in large cities.

Assuming the distant amateurs desire disconnection from the link (or the calling station desires distant disconnection) another 3 digit code will "bring down" that transceiver (example: 128). Data packets may be moved either to the distant repeater, or left in the electronic mailbox as required. Continuing overall system capabilities one step further, we can use tone control and port-located microprocessors for handling frequency offsets and Spread Spectrum hopping sequences. This capability would permit an individual amateur operating on 146.52 mHz to "catch" the network's scanning transceiver, establish different network - relayed frequencies, ard proceed in the previously described manner (Example: 146.52/.52 into network, 146.16 out of network. One or two fully microprocessor-controlled transceivers required for this option). A full description of the network would, obviously, encompass numerous pages of discussion. We thus leave those operations open for your investigation and expansion. The network outlined is a coarse system for future communications techniques. We hope this "first basic step" will inspire unlimited expansions.

DEAF TELECOMMUNICATIONS NETWORKING

Barry Strassler
Executive Director
Telecommunications for the Deaf, Inc.
814 Thayer Avenue
Silver Spring, Maryland 20910

Thank you every one of you for giving me the opportunity to give you a glimpse into the world of deaf telecommunications. It should be of interest to you because our technology interrelates with yours in many ways, and perhaps might even spur you on to better developments.

Life is full of historical accidents that shape the course of history. The particular historical accident that I wish to bring up refers to one of your people, a deaf one at that. In 1964, Robert Weitbrecht, of California who is one of the world's few deaf licensed radio hams, was hiking in the mountains with a group of friends. Another group of hikers converged onto Weitbrecht's group. In that second group was a couple whose son is deaf. That couple overheard a very distinct speech and knew that it was from a deaf man. This couple walked over to Weitbrecht and introduced themselves to him. In the course of conversation, this couple learned of Weitbrecht's diverse interests, one of which was radio-TTY. Coming home, this couple related this encounter to their deaf acquaintance. This acquaintance knew of a deaf dentist who had wished for some sort of telephone communications tool to be able to reach his clients. One thing led to another, and introductions were made all around. This deaf dentist inspired this deaf radio ham/hiker to come up with such a TTY device for the deaf. This is how history was made.

Our present TTY or better known as TDD (Telecommunications Device for the Deaf) network consists of 75,000 installations all over this country. This phenomenon, starting slowly in 1968 and growing by leaps and bounds each year, is not without its curses as well as its blessings. The blessings we all know about. But the curse is the ASCII/Baudot controversy. In 1964 when Weitbrecht came up with his prototype TTY modem, it was designed to work with then-current teletypewriters using the Baudot code. He chose this code for convenient reasons which were valid at that time -- availability of surplus Teletype Corporation machines by communications carriers, ASCII technology being so new and not fully understood, ASCII parts and equipment being so costly and way above the means of the average deaf households. At any rate, we are witnessing a spectacle -- the 75,000 TDD units, all of them predicated on the Baudot code, being considered very useful but very obsolete when we look at the ASCII tide engulfing us today. We should switch to the ASCII code, but it just: cannot take place over night. This would suddenly disenfranchise those deaf TDD users, many of them from low-income households.

A way to bridge this ASCII/Baudot gap is a dual modem capacity. The average TDD has a life expectancy of some five years. So it is in the replacement dual-modem TDDs that we hope will slowly swing the pendulum towards the ASCII camp. Protocols are another matter. If I had made this speech a year ago, I would not be so sure o:f protocols. You see, when a flashing light indicates a telephone ring in the home of the deaf, we would have to determine whether it is a voice, Baudot or ASCII call. It would be cumbersome to fiddle around with full duplex/half duplex and with originate/answer switches. But today, technology has come up with an automatic ASCII/Baudot detector, so this set of protocols is taken care of.

Now we are into electronic mail systems. There are several such systems which have served the deaf. One of them is DEAFNET which serves the Washington, DC and the San Francisco Bay deaf communities. It has ASCII/Baudot capabilities. Another one was the Hermes System which served the Boston area before funding ran out. But replacing the hermes is the GTE Telemail System. The Hermes was self-containing in that it was restricted for the use of the deaf only, and ASCII terminals had to be used. Now they have the GTE Telemail, which is a piggy-back system. Here, in. addition to DEAFNET, we have the Virginia TTY message system which is totally Baudot. All this is very promising for us in the years to come.

We have been asked by many radio hams about the possibility of the deaf getting involved with radio-TTY, This is another possibility, but not without *problems that must be overcome. One is attitude -- the deaf like to ragchew with another deaf acquaintance; because there are so few licensed deaf hams around, this is a factor. Second is the matter of taking Morse code exams. There is always that hearing impairment will thwart the mastering of the Morse code. Perhaps this is psychological, but it still must be overcome, Third is the lack of familiarity with the various kinds of radio equipment, such as short wave, Citizens Band, pagers and the like. We, the deaf, look on all of these as one thing, and it seems very perplexing.

I personally feel that if my organization and your organizations work together on some kind of educational campaign to interest deaf technological enthusiasts in the ham radio field, then progress can be made. I travel a lot and am forever besieged by frustrated deaf radio-hams-to-be. So the interest is there even on a small scale.

There is much more to the future of communications for the deaf aside from the telephone, the computer and the radio, Your world is laden with advanced communications devices. With ingenuity being present, it would be remarkable that every kind of device that you use for communications purposes will have its for-deaf modifications. This is our dream now a.nd in the years to come.

FROM RTTY TO PACKETS

Joe Kasser, G3ZCZ

Introduction

Putting a microprocessor into an RTTY station can improve the usefulness of the station by orders magnitude. RTTY comes in various aspects: there is the old fashioned Baudot network chugging along at 45.5 bauds; the new ASCII links are running at 300 bauds and exchanging computer data; and, up and coming are the packet radio networks. This article discusses the whole arena of digital communications and shows how each is a step in the whole picture; and how as computers are added,, a digital network can be developed that can accommodate users with almost any equipment so that radio amateurs having incompatible equipment (e.g., a Model 15 Baudot machine and an ASCII terminal) can communicate via microprocessor based repeaters,

Digital Communications

Digital communications at this time refer to Morse code or RTTY communications. Morse code is a digital communications medium in which the presence or absence of a signal and the spacing of the signals define the content of the data. RTTY communications use either Baudot or ASCII codes to relay written information which is displayed by a radio teletypewriter or, as is becoming more evident, a cathode ray tube terminal. 'This section deals with RTTY communications at vhf/uhf.

Vhf/uhf RTTY is usually transmitted using audio frequency shift keying (afsk) on fm equipment.

Transmissions can be asynchronous random length using ASCII or Baudot codes, or fixed--format packets of data using ASCII or some other 8-bit word code, Putting microcomputers into the communications link can allow anybody equipped with digital communications hardware to communicate with anybody else also equipped with digital communications hardware, even if their equipment cannot communicate directly. Thus, G3ZCZ/W3 who has ASCII NO-baud equipment could communicate with WA3LOS who is equipped with a Baudot Model 15 teletypewriter. This is an ideal way to provide for low-cost, low-speed communications at advanced levels. This paper examines various aspects of digital communications.

RTTY Repeaters

The usual RTTY repeater usually provides coverage of a large metropolitan area. The frequencies of 146.10 MHz (input) - 146.70 MHz (output) have been assigned to such repeaters in the USA, although often other frequencies are used. Simple repeaters receive the afsk tones and reradiate them directly just as if they were audio signals in a conventional repeater, More advanced repeaters demodulate and then regenerate the signals.

RTTY repeaters first came into operation for the same reasons that audio repeaters were utilized, They can provide an extended coverage area as shown in Fig 1. Thus in the coverage area, anyone having suitable equipment can copy signals on the frequency.

RTTY, however, is a slow mode of communicating. Most people cannot even type at 45.5 bauds, and even when sending pre-stored messages at full machine speed, messages still take a long time to send. A two-way RTTY contact can take an hour or so to pass information that can be passed in minutes by voice. RTTY does have one major advantage over voice communications, however, and that is unattended operation or autostart capability.

RTTY Network

The RTTY network works as follows. All stations monitor the same frequency, either at hf or vhf., Messages are sent blind: that is, when a message is originated into the network, the sender does not know for certain if the destination station is monitoring the frequency, unless contact is first established. In the evening, or at weekends, this may not pose much of a problem because the probability of someone being at home is great. However, by day, that probability decreases, Thus, if contact cannot be established directly, the message can still be sent, but there is the probability that the destination will not be on line, and it will be lost,

If the message can be stored in a central computer by the sender, and then retrieved later by the receiver, the probability of successful transmission of the message from sender to receiver becomes a certainty. The addition of a computer thus becomes an asset to the network,

If several stations in the network have computers capable of answer back, the utilization of the computer may be reduced, A sender can put out a direct call, If an answer is not received (indicating that the destination is not on line or monitoring at the time), the message can either be transmitted to the computer for storage or held and transmission retried at a later time, It is also possible for the network computer performing the store-and-forward operation to rotate among the various member station computers on an as *available* basis as

long as the network computer has a **distinctive** identification.

Some repeaters allow ASCII signals at data rates of 300-1200 bauds to be carried. Stations with Baudot equipment cannot directly communicate with stations having ASCII equipment. The network computer can contain a **conversion** capability wherein messages received on one mode would be converted to and retransmitted using the other mode.

In use, ASCII messages would be relayed directly. The ***high-speed*** ASCII message being input would be stored in a memory buffer and an output program would transmit the contents of the buffer at **45.5** bauds even while the buffer is filling up at the ASCII data rate. Incoming Baudot messages can either be transmitted directly in ASCII at the higher baud rate (but still at a real character spacing of **45.5** bauds) or can be stored and then transmitted later as a single message at full speed. In the latter **case,** some sort of tone or signal would have to be placed on the output frequency to notify all users that an incoming signal is present and is being stored but not transmitted. It would probably be better in this case to retransmit the Baudot message as it is received and then follow it with the ASCII message upon completion of the reception.

In use, the operator at his station types up and transmits a message. The message is transmitted directly to the target station or is stored in the computer. Some time later, the operator will check to see if a reply has been received. Depending on the degree of sophistication of the network, he may even be able to interrogate the network computer to see if the message has been forwarded. **Thus,** the concept of store ***and*** forward in the network computer is really a logical extension of autostart techniques,

These techniques for Baudot/ASCII conversions allow amateurs equipped for different modes of operation to communicate. The scheme presented above does suffer from the limitation that only one message can be transmitted at any one time,

The Baudot network can **be** classified in the ***dumb*** range, The users of this network are usually operating in the manual mode, possibly using paper or audio tape to facilitate operations, Incoming messages are printed out and possibly punched on tape, Very little selectivity exists to separate messages addressed to a station from others on the frequency, (A few hard-wired selective calling units do exist.) Error detection and correction techniques are minimal,

The ASCII network can be classified as a semi-***smart*** network. **Most** users have some kind of microcomputer-based system. Communications are at **300-**9600 bauds, but again have a minimal amount of error-correction facilities. This network can be used to transfer files between computers and in fact is being used as such.

The packet network can be classified as a ***smart*** network since *error* detecting and correcting techniques *can be* used. Thus, if the receiving station detects an error in the message, it can automatically request a retransmission of the bad message to ensure that the traffic is correct.

Burst Mode **Communications**

Consider a digital repeater operating at 1200-9600 baud ASCII. Each user has a small **microprocessor-**based terminal that contains a minimal amount of hardware and software to perform the following operations:

1) store a few lines of **text;**
2) remember who the message is going to;
3) remember the call sign o:f the station; and,
4) display incoming and outgoing message.

These capabilities are not too advanced on current smart dedicated microprocessor-based **RTTY** terminals.

In use, any amateur would start typing a message at the terminal, When a line of text has been input, the microcomputer would check the frequency to see that it was clear and then transmit that line of text at the high-speed rate (verifying it on the repeater output frequency to ensure that it was reradiated properly), The amateur typing away at the terminal need not even know when the transmission burst is sent. Since most people type slowly compared to 1200-9600 bauds, the terminal will spend most of its time in the non-transmitting state, **Thus,** a number of amateurs could be using it at ***the*** same time, Anyone monitoring the channel would pick up all the signals, Each line of text could belong to differing messages and thus would appear **to** be garbled, If, however, each line of text was prefixed by the call sign of the target station (and suffixed by the call sign of the sending station) and the microprocessor in each terminal was programmed to respond to and display messages only addressed to its call sign, the traffic on frequency would become invisible and this ***time*** sharing of the repeater would be unnoticed by the users, in that incoming messages would be displayed a line at a time instead of a character at a time as in the conventional network. These lines of text that are transmitted in a. burst mode can be called packets of data.

Once a microprocessor is put into use **in** storing the input characters and then bursting them out as a packet, it can also be used to provide some error-detection capabilities.

Packets

A packet of data can be considered as a **high-**speed burst of information. The typical RTTY frequency can be **occupied** only by one QSO at a time, and data is sent at the rate that it is typed. Thus, although a Baudot network can pass data at 60 words per minute **(wpm)** using conventional mechanical teletypewriters, a real data throughput of 60 wpm is

only achieved when running at machine speed. Since **the** actual typing speed in a contact varies as a function of the digital dexterity of the operator, the data throughput is slow. Computers can be used to speed up the flow of information and improve the channel occupancy.

Suppose the data being typed is buffered by the **computer.** The contents of the buffer can then be output at high speed (say 1200-9600 bauds) as a burst. If the computer checks that the channel is unoccupied before transmitting,, there will be a minimal amount of loss of data due to interference (two stations transmitting simultaneous bursts are the only practical cause of such interference). If each packet or burst was prefixed by the call sign of the destination station, it would be uniquely identified, The computer at the receiving station would ignore all bursts addressed to other stations. Thus, many **QSOs** could take place timesharing the channel, An example of such a scheme is shown in Fig. 2, Any station could display all information relayed or just the messages addressed to itself. Thus, the addition of a minimal amount of software would improve the use of the basic RTTY repeater network,

Once computers are used for high-speed data burst **communication** links, advantage may be taken of the capabilities of the computer to provide error checking and correction capacity, Thus, protocols can be defined and adopted with those ends in mind,

The main problem here is that new stations joining the network can ***bomb*** it if their equipment (hardware or software) is not working correctly, If an average of one new station per week joins the network and bombs it for two evenings each time, the network will suffer a lot of down time.

Several techniques can be used to minimize this problem. The station software can be tested out on a simplex or different channel, or a cheap **special-**purpose microprocessor-based circuit card could be developed that would act as a front-end processor fitting between the computer and terminal unit. It would contain the buffers and network communication algorithm. Anyone wishing to access the network would be required to obtain the unit in a similar manner to the way that a tone burst or sub-audible tone is required for access to a large number of two meter (audio) fm repeaters. The front-end processor card could be mass produced at low cost once protocols are established. If designed properly, the protocols could be PROM-based and the same unit could be used for a number of different protocols by plugging in a different PROM for each protocol in the **likely** event that different protocols be established in different networks.

The actual protocol provides a means for ensuring error free transmission of a message and is transparent as far as the message itself is concerned.

The analogy in conventional amateur radio is that the sounds emerging from a loudspeaker at the receiving station are the same as those entering the microphone at the transmitting station, In an interference free situation it does not matter to those sounds if the modulation technique was am, fm, ssb, or **dsb**.

The Packet Network

The packet network is set up for stations who can communicate directly using packet techniques. **The** advantages of packet communications are many and include the timesharing of the channel, relatively high speeds and error detection and correction,

The block diagram of a packet network would be identical to an RTTY Baudot or ASCII network, but packet transmissions offer one big advantage in that a packet repeater can operate in the simplex or single-frequency mode. In this network, all stations monitor the same frequency. All stations can transmit to and receive from the central store-and-forward station (repeater). In use, a station would store the message as received. It would then transmit the message on the same frequency so that the intended recipient would be able to receive it, If the intended recipient was not able to copy the original message, it would be able to detect that it had received the same message twice because the repeater would have set a flag byte in the message header indicating that the packet was a relayed version.

The conventional repeater requirement for two frequencies (input and output) at one time has now been replaced by the requirement for two time frames (original (input) and retransmitted (output)) on one frequency,

Network Communications Language

The Baudot and ASCII RTTY networks require some routing signals to ensure that messages are routed to their intended destinations. A suitable source for these signals is the *Q* Code commonly used by radio amateurs., The use of slightly modified Q Code signals will make the messages easily readable by both man and machine,

For example, a message such as

WR3ABU :QSP: **WA3VXE** :QSO: ALAN

PLEASE CALL ME ON THE TELEPHONE AFTER NINE TONIGHT **:** QSL: DE **G3ZCZ/W3**

is almost already understandable even without explaining that

:QSP: means (please) relay to call sign following

:QSO: the message following

:QSL: end of message/confirm reception,

In other words, the store-and-forward computer at WR3ABU was asked to forward (QSP) a message to WA3VXE and confirm its reception by G3ZCZ/W3. Later on when WA3VXE signs in to the network, he would send

WR3ABU :QRU: DE WA3VXE

which means WR3ABU do you have any messages for me.

WR3ABU would reply

WA3VXE :QRU: G3ZCZ/W3, WB2YUX/3, DE WR3ABU meaning that there are messages from G3ZCZ/W3 and WB2YUX/3.

WA3VXE would then send either

WR3ABU :QUA: DE WA3VXE

or

WR3ABU :QBM: G3ZCZ/W3 DE WA3VXE.

If you know the Q Code, you will know that QUA means *send me* all new messages, and QBM means *send me the message from "----"*.

Other examples are:

WR3ABU :QRT: DE G3ZCZ/W3

which signs G3ZCZ/W3 off the network.

WA3VXE :QRL: DE G3ZCZ/W3

which asks WA3VXE if he is busy, No response within a short period of time means that he is not there. If he is, the replay would be

G3ZCZ/W3 :QRU: DE WA3VXE

i.e., an automatic answer back.

Note that WR3ABU would not respond to the QRU because its call sign was not recognized, G3ZCZ/W3 would then send his message as follows:

WA3VXE :QSO: ALAN, IT LOOKS LIKE WR3ABU IS DOWN, SO I TRIED YOU DIRECT :QSL: DE G3ZCZ/W3.

The response would come in a flash (or at least at 60 wpm)

G3ZCZ/W3 :QSL: DE WA3VXE

Hence, even if WR3ABU was monitoring the transmission and recognized its call in the text, since the call sign was not immediately followed by the :Q sequence, it would forget that it had just recognized its call and go back to sleep.

A message in the form:

WR3ABU :QSP: GB3LO :QSP: G8BTB :QSO: PAT ARRIVING ON THURSDAY 22 JUNE :QSL: DE G3ZCZ/W3.

Would instruct WR3ABU that a message is to be sent to GB3LO who will then forward it to G8BTB. This extension assumes that GB3LO is the store-and-forward computer in a second network in which G8BTB is operating.

The ":" placed before and after the three letter group of the Q Code makes recognition and decoding easier since all control language statements begin with :Q and a : is the fifth character in the sequence. An example of some of the Q codes suitable for use in the dumb and semi-smart networks are shown in Fig. 3.

Amateurs using Baudot equipment would have to type the control language statements in full. Those having microcomputers could type ASCII control characters which would be software converted to the equivalent S-letter control group,,

Network Control Language

The Network Control Language (NCL) provides the computers with information as to what is to be done with the data in a message. Numerous languages exist to provide computers with instructions, but few exist for communications purposes. NCL is written in some other language and is not a true language as such, but is an implementation of an NCL Program in which the man-machine (or machine-machine) dialog is in specific format. Most radio amateurs are familiar with the Q code. Words such as QRM, QSL or QSO are understood by them all. Others, such as QRA or QSP, may not be understood unless the radio amateur is used to traffic handling; but, since they already have some knowledge of the Q code and--better still--an idea of the concept behind it, the Q code is an ideal *language* f'or telling the computer how to route or process data.

The NCL based on the Q code can be used at all levels of digital networking, starting with a lowly Baudot circuit all the way up to a packet network carrying video as well as audio packets of data. Of course, the packet network with its fixed length packets can simplify the actual transfer of data by using positions in the packet to convey information, Thus, the call sign of the sending or receiving station could always be placed in a certain position in the packet, rather than--as in the random length RTTY message--use the Q code to specify originator or destination.

NCL is used to communicate with the communications software in the computer ox in a stand-alone packet terminal interface. Apart from the use of the : as prefix and suffix for the control word (example; :QRM:) the Q code can be used pretty much in the conventional meanings. Thus, the Q code shown in Fig. 3 can be converted to NCL as shown in Fig. 4, The call signs can also be expanded upon, drawing upon the usage of *wild card* characters used in several microcomputer operating systems.

Theme wild card characters are known as general call characters and allow the sender to send a message to a category of stations.

The "?" character may be used to match any single character, or number, For example, G3Z?? matches any call sign in the G3ZAA-G3ZZZ series. W1??? matches any W1 call with a three letter suffix. The "*" character matches any section of a call sign (including a null character) as follows:

G3* matches anybody with the G3 prefix

G** matches anybody with the G prefix

GM3* matches all calls with the GM3 prefix

3 matches any call with the three digit in it

*** matches any call in the world

'The two general call characters can be mixed at will, For example, G*3A?? will match any call in the United Kingdom in the G3AAA-G3AZZ series, including those with the GM, GW, GC, GJ, GU, and GB prefixes. Thus, GB3AAA, GM3ART and GW3AAA would be matched. Note that G3AA would not match because three ? characters were used.

Network Implementation

RTTY, packets, ASCII, how do they interconnect? How does an amateur who only has a Model 15 RTTY machine communicate with an amateur who has a packet terminal? Should he? In the conventional communication modes an amateur who only has a Morse Code (cw) station can communicate with another amateur who is using voice (ssb), but neither of these can communicate with someone else using the teletypewriter (RTTY). There are many Baudot stations in existence; newer amateurs may come on the air with ASCII using microcomputers and advanced amateurs can use packet techniques. In general or random communications, in which one amateur calls CQ and wants to see who (what?) comes back, all modes usually work other stations equipped with the same mode, Thus, Baudot RTTY stations have established frequencies within the amateur bands where they have the greatest probability of finding others suitably equipped. It is conceivable that ASCII and packet stations could do the same. The big advantage of packets and computers in the RTTY area is that delivery of messages can be guaranteed, and by using a hierarchy of rf links, messages can be relayed between amateurs having different digital equipment. Thus, a Baudot station could send a message to an ASCII station.

Consider the hierarchy involved: many local area nets exist using Baudot equipment. These nets may be on vhf or on hf. Each mode has its advantages and disadvantages. Fig, 5 shows a potential situation for interconnecting such a network with a new ASCII network in the same local area. In its simplest implementation, two repeaters are co-sited, One is a conventional Baudot RTTY repeater, the second an ASCII repeater, In the normal mode the two are separate; ASCII stations talk to ASCII stations at 300 bauds or even at 1200 bauds and Baudot stations talk to Baudot stations at 45.5 bauds,

However, by using a translator, Baudot stations can communicate with ASCII stations, since the translator will perform the code/speed conversions from one to the other, The translator can in a sense be thought of as a third repeater,

Each network can operate independently, When somebody on one network wishes to send a message to someone on the other, he can use the network control language based on the Q code to instruct the translator accordingly. The translator should thus be able to store the message for later forwarding in case the other network is carrying traffic at the time that the message is originated, The translator will transmit the message later when the other network is free, or can hold it until the intended recipient signs in (on either network) and requests his message.

The translator can be used to perform the store-and-forward function for both networks at the same time, Once computers are put into the network, they can perform as many or as few tasks as their owners desire, The different functions can be split up between computers, or all the computers in the network can have the capability to perform all the tasks, thus providing a high degree of redundancy and reliability in operation,

Network Hierarchies

The lowest level is the 45.5-baud Baudot network, Baudot machines are usually available for less than $50 at local hamfests. They can be Large and noisy, but they work and can easily be interfaced to a computer. They can thus be used in basic or conventional RTTY stations and then when a computer is incorporated in the station, can be used as a hard-copy printout device or even as a system console. Since thousands are already in use, they should not be obsoleted just because newer and better things such as 1200-baud ASCII are now available. Their limitations will soon become apparent to the user; who will then upgrade to the newer and better devices, passing the Baudot equipment to someone else, allowing them to join in the fun. Thus, Baudot operators will still be able to enjoy simplex contacts at hf and uhf,

The next level is the ASCII user, Here the baud rates can go as high as 9600 baud and yet remain within a 3 kHz audio bandwidth. Common tone pairs used by amateurs in the USA are Bell 103 tones for 300/110 baud contacts and Bell 202 tones for 1200 bauds. It is thus possible to build a digital repeater that can monitor the incoming signal and perform conversion to a different code/tone pair for retransmission on the output. An example of such a device is shown in Fig 6. The incoming signals are demodulated and converted to

serial data by the different receive terminal units. These **are** fed to a microprocessor module **via** serial ports. The microprocessor is operating as a dedicated controller in this environment. Under normal conditions, the microprocessor retransmits the signals **in** the same format as they **were** received. Thus, if **45.5-baud** Baudot signals **were** received, that is what would be transmitted. If, however, the message is prefixed by a control code, the microprocessor would cause the signals to be transmitted using **a** different modulation technique. The microprocessor would also perform speed conversions as well as time conversions and provide first-in, first-out (FIFO) buffer **in** the event that the retransmitted signals were at a slower baud rate than the input signals. With this kind of arrangement, amateurs with **1200**-baud ASCII terminals will be able to communicate with **other** amateurs having Model 15s or similar Baudot equipment. The limitations of 45.5 bauds, as compared to the high speeds, will soon cause a decrease in the number of stations using Baudot and **a** corresponding increase in the number of ASCII operators. This approach, however, does allow the newcomer to join in with a minimal investment, encourages **upgrading** of equipment (by example), and does not penalize those with older **equipment.**

The basic data link allows errors to creep into the message. Errors **are** caused by noise or interference entering the communications path. In most of **today's** amateur digital communications the occurrence of errors are not too serious and can easily be detected by visually scanning or reading the received text, **When** computers are doing the communications, they also **have** to have a means to detect that errors have occurred in the transmission of the message.

Linking of Networks

The discussion so far has limited itself to single area networks serving a common set of users. The **requirement** to **interlink** networks exists. Being radio amateurs, the interlinking technology will of course be by radio. **An** example of interlinked area networks is shown in *Fig.* 7. In the figure, each network has at least two links to the outside world. The link may be through the central store-and-forward computer or it may be through one of the users. For example, if it is used to link two networks, the link may be either via the central computer or via the user, but if vhf/uhf is used, the link would probably be via **a** user who is located between the **two** networks and is able to access both directly (possibly only with the assistance of high power and directional antenna).

The communications on the interlinks use the packet mode. This is because the links will probably have lower signal-to-noise ratios than the vhf/uhf local network paths and the probability of interferences is greater.

Conventional vhf links also suffer from a **routing** problem. How does a message get from network G (in Fig. **7)** to network A?

Does the originator specify the routing, the final network store-and-forward call or nothing at all? **Which algorithm** is to be used **- fixed** path **-** random, etc? Professional networks have spent thousands of dollars on this problem. Radio amateurs **also** cannot guarantee that **all** the links will be operational with a reliability of 0.999. What happens when **a** network node goes down for **a** while **-** are messages going to get backed up, are they going to get lost?

Hf propogation also suffers from its own characteristics, The ionosphere that reflects back the hf radio signals is **a** dynamic medium. Its properties change from minute to minute, are different during the day and the night and are affected by solar activity which may enhance or detract from the reflecting properties of the **inonosphere** at any particular frequency. Thus, the situation shown in Fig. 8 is typical of the conditions under which radio amateurs operate. Stations A and B are in direct contact with each other. Station C can also hear A but cannot hear B. If station B is transmitting, Station A will be silent. When Station A is silent, Station C may try to send a packet to Station A and interfere with the packet that Station B is sending. Station **D,** who cannot hear any of them at this time, **may** transmit to someone else and as conditions change will interfere with A, B, or C. This situation is not impossible, it is just **difficult** to design around.

Several techniques have been developed to minimize the QRM situation, Each station in the network can transmit **at** random. If a collision between two packets **occurs,** i.e., one interferes with the other, the receiving station will not be able to send an acknowledgement to the sending stations so the sending station **will** try again later. If each station waits a different random amount of time before transmitting its packet, there is a good probability that the second time that **a** packet is transmitted it will get through.

Another alternative is to give each station **a** fixed time slot for transmission. Thus, Station A would always transmit during the first second of any minute, Station B during the second, and so on, If the stations are referenced to **WWV** or any other standard frequency and time transmission, a minimal amount of interference will occur, but the throughput will go down since **a** station may have no packets to send but that time slot will still be reserved for it.

Adding the interference problem to the routing problems, hf networks are in themselves a problem.

One solution could be to use a random transmission sequence based on the probability of **a** successful contact, This means that messages are only originated if there is a good probability that **there** will be propagation to the destination

or target station at that time of day.

Given that a system in which everybody cannot hear everybody else, in which propagation is uncertain is a difficult system to operate, it follows that the converse is true - i.e., a system in which everybody can hear everybody else, in which propagation is 100% predictable is ideal. This situation occurs if a communications satellite can be utilized as the relaying medium.

Fig. **9a** shows the same four stations now using a communications satellite to relay messages, They can all hear each other, and since the orbit of the satellite is known, they can compute the time when propagation will be possible between any of them. If the **AMSAT** Phase III or Phase IV satellites are used, each covering large areas of the world, a global network takes on the shape of a local network as sketched in Fig, **9b.**

The satellite itself does not contain any **store-**and-forward equipment. The gateway stations on the ground each act as a local user to the satellite, They can all monitor the frequencies so can pick up any traffic targeted at themselves. Since the satellite operates in a duplex mode, they can all monitor the **downlink** when uplinking and can detect errors due to noise, or due to collisions and take appropriate steps. Since the orbit is known, they can determine mutual visibility and store messages until the target comes into a mutual visibility window.

Each gateway station may act as the central store-and-forward station or as one of the regulars in its own network, and as long as the gateway is operational any station on the network has access to the network as a whole, Thus, it can truly be stated that ***the sky's the*** limit ***in amateur radio digital communications.***

Using the Networks

The **RTTY** networks can be used in an identical manner to existing RTTY channels, It does not matter if they are Baudot or **ASCII,** CQ random, or **point-to-**point (autostart). Communications **take** place in a conventional **manner.** The use of NCL only becomes necessary if a message is to be stored in or retrieved from a computer, The location of the computer also does not matter, The user of the packet network will usually have a dual processor system, as shown in Fig, **10.** The Terminal Interface Program (TIP) may or may not be part of the main computer. The TIP can operate in two modes; monitor or terminal, In the monitor mode, it can pass every packet it receives to the main computer. The destination of the packet does not matter) this mode is a good debug mode for testing the TIP, as well as providing a level of confidence in the early days. Since the packets may **or** may not be complete messages in themselves, the output of the TIP may or may not make **sense.** In the terminal mode, messages only addressed to the user will be output by the TIP,

If the **TIP** is a stand-alone board, having an RS-232-C interface, it can be connected to a terminal device and used as a dumb packet terminal, A dumb packet terminal is a terminal that can send and receive packets,, It contains the basic **low-**level software to format a packet for transmission, and acknowledges reception of, and unformats, a received packet, Such a stand-alone board, microprocessor based, is a low--cost introduction to packet techniques, There is, of course, no reason why the TIP function could not be performed in software by the host machine, apart from the obvious one that it may tend to prohibit the use of the computer for other purposes. An outline of a stand-alone TIP is shown in Fig. 11, The breakdown shown for the TIP comprises a standard microcomputer, The control program is in PROM, the data storage area is in **RAM.** The more RAM that is available, the greater the length of or number of packets that can be stored in the TIP.

It is **envisioned** that the user will graduate from the monitor **mode** to the terminal mode pretty quickly. After the novelty of receiving packet transmissions has worn off, the unit will be switched to the terminal mode, Of course, the user may temporarily revert to the monitor mode at any time to check that the TIP is still operating after a long period in which no messages have been received had occurred,, The user, via the terminal, or the host computer can communicate with the TIP by using **NCL.** In this way, the user does not really care about the mechanics of **getting** messages across. All he is interested in is the message, **i.e.,** the high level protocols, The low-level protocols of exactly how and when a TIP goes on the air can be left to the minority of technical hackers amidst the ranks,,

It is desirable that the same software be used in all the **TIPs. The** real world will, however, not be the same as the ideal world, The standard **PROMs** supplied with each TIP could be programmed with the station call sign as *****.** These general call characters will respond to all call sign addresses which is the monitor mode situation, Thus, in use--at power up--the TIP would output a sign on message to the serial plot such as AMICOM TIP REV 3.6 QRA which would identify the network program protocol and the revision level, The TIP would then be in the monitor mode and the user would change to the terminal mode by entering :QRA: followed by the station call sign (including general call characters **(i.e., :QRA: G3ZCZ).** Note that a call sign such as **G3ZCZ/4X** would be recognized as having the 4X prefix--not the, or as well as the, G3 prefix,

One advantage of a separate TIP is that it tends to maintain the integrity of the network. Consider a network in which one new user a day joins in, Given the number of radio amateurs and the number **computers** in existence, that is not an unreasonable assumption, If each user has to bring up software and hardware at the same time in an area in which he has not worked in before, the probability of errors occurring, bombing or typing up the

network, is high. The network could thus suffer from a lot of down time due to those newcomers not quite being able to access. If a standard board is available, the software can be provided to drive the integrated circuits on the TIP which will minimize the number of bad signals on the network frequency, If NCL is used to control the operation of the TIP, the TIP can be driven by any computer having a serial port, programmed in any language.

Once messages begin to flow into and out of the TIP, some high-level control is desired. This high-level control forms the user interface to the network, not at rf, as in the case of the simple RTTY network. Again, here a hierarchy is possible, The TIP can drive a simple terminal in the monitor mode or can interface a microcomputer with floppy disc capabilities for storing (and forwarding) messages.

In a simple RTTY network, only one contact is taking place at a time, irrespective of how many stations are taking part in the round table. In a packet network, only one packet is being transmitted at a time, but successive packets as received at one station need not be part of the same message. There is, thus, no reason why when two stations are in contact, packets originated by other stations could not start appearing in the network and some of those packets could be addressed to one or both of the stations already in contact.

Consider for a moment the working of the TIP in its receive mode. The program must monitor the frequency and read the leaders of all the packets on the channel. Messages intended for the TIP's own station will be output at the serial port. What happens if the TIP has received a packet from one station, but that the packet was not a complete message? While the TIP is waiting for the next packet that will contain more of the message, a packet arrives addressed to the TIP but originated from Station B. Does the TIP reject it because the new packet will garble the message currently being received?

If the TIP rejects it (by not acknowledging it), Station B may keep trying, thus slowing down the communications speed of the channel because its packets will be ignored by our TIP, yet are using up time that could be utilized by Station A to complete his message or by other stations to pass different traffic. If the TIP does not reject it, a computer (either the TIP or the host) has to have some way of recognizing the different origin of the packet and storing it in the relevant bit bucket. On the other hand, the TIP could send a busy packet to Station B and anybody else which means that they can either try again later, or wait until they are called back. The busy packet could contain a flag bit or byte or NCL word that instructs the calling station as to which of the choices to follow, The *call again later* technique requires minimal software in the receiving program, but can increase the amount of traffic, as the calling station keeps trying for a contact and keeps getting *try again later* responses. The *wait for me* to *call you* response requires some additional software in the receiving program to store a list of stations to call and notify that the TIP is ready for a new message, There the tradeoff is TIP software complexity against network traffic load,

Most disc-based BASICs (and other languages) have the capability to have more than one disc file open at a time. It is, thus, logical that a high-level protocol can be used when the TIP is used together with a host computer to allocate space on a disc for more than one incoming message at a time.

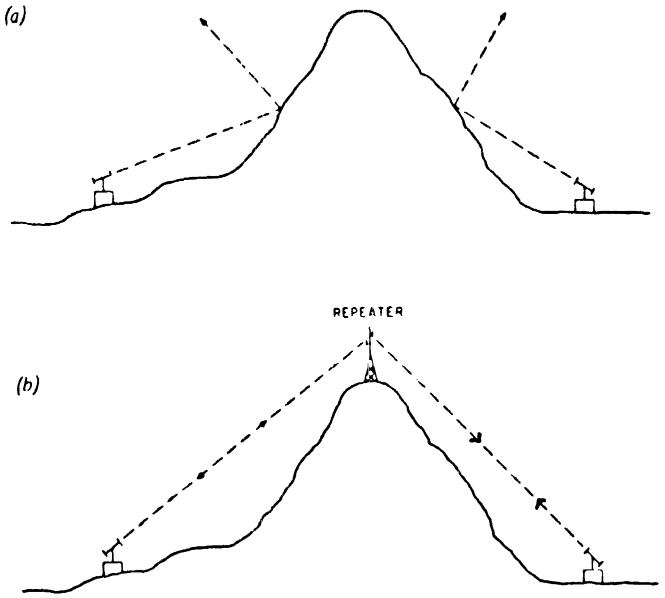
The user being interested only in the whole message does not care how many packets it took to receive the message. The network manager, on the other hand, may have different interests and a distinction should be drawn between the requirements of the network user and the network manager, Once a whole message has been received, the computer can signal its owner accordingly. The multi-message arrival problem is also significant in the situation in a store-and-forward mode, where the packets are assembled into complete messages for storage purposes.

Another situation that has to be looked into is the CQ call or rather the response to the CQ call situation. One of the major advantages of packet communications is unattended operation, A terminal can thus be programmed to respond to a CQ call so that a newcomer joining the network will have a response to his initial call. What happens if 20 stations or so try and respond to the CQ call? If multiple simultaneous received messages are allowable, a station can end up having several simultaneous QSOs. An extreme example of this condition is the effect on the network due to the appearance of a rare DX station. Consider what could happen to the network if a rare DX station signs in or originates a message.

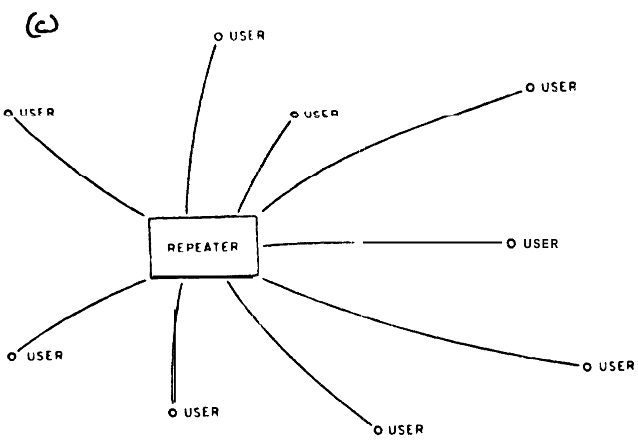
Avid DX chasers spend a lot of time and money on working new countries. It is thus very likely that these avid DX chasers could leave their TIPs in the monitor mode and have custom DX capture software in their host computer. Thus, the appearance of a DX station can be detected if it originates even one packet into the network. The result could be a pileup. Each of the DX chasers will originate packets aimed at the DX station. Collisions will occur, due to the QRM, calls will be tried over and over again and the network will be tied up for a long time even if the DX station has gone QRT (possibly due to front-end overloading of his TIP software?) as the DX chasers keep trying for a message acknowledgement. Thus, packet communications software deemed workable with few stations (initial net situation) may be less than optimum in a wide area established operational network.

Packet communications offer a revolutionary new means of passing communications to amateur radio. For optimum results, it is very advisable that the low-level communication protocols and the high-level software be well thought out, flexible, and easily adaptable to changing circumstances.

Fig. 1. Extending Communications Range by Use of a Repeater

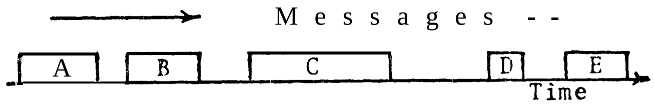


a shows that if there is a hill between two local stations, it is not possible for them to communicate by means of VHF. If a repeater station is positioned at the top of the hill as in b, communication becomes possible.



A city-wide radioteletypewriter VHF repeater link.

Fig, 2. Timesharing of a Communications Channel



Message A is sent by G3ZCZ/W3 to WA3LOS

B is sent by WB4JFI to W3ZM

C is sent by G3ZCZ/W3 to WA3LOS

D is sent by W3IWI to K1HTV/3

E is sent by WA3LOS to G3ZCZ/W3

With proper prefixes G3ZCZ/W3 and WA3LOS will not notice that the channel. is being shared by others during the times that they are not actually transmitting anything.

Fig. 3. Extracts from the Q Code.

Code	Question	Answer or Advice
QRA	What is your identification or call sign?	My call sign is . . . .
QRL	Are you busy?	Yes. I am in use by . . .
QRM	–	Your transmissions are being interfered with.
QRQ	Shall I speed up to . . . . bauds?	Yes.
QRR	Are you equipped for automatic operation?	Yes.
QRS	Shall I slow down to. . . , bauds?	Yes.
QRT		Signing off (log off).
QRU	Have you any messages for me?	Yes, messages are from. . . .
QRV	Are you ready?	Yes.
QRX	Will you wait?	Yes.
QRY	What is my turn?	Your turn is number .
QSG		Send . . . . messages.
QSK	Can you operate full duplex?	Yes
QSL	Will you confirm?	Confirmed.
QSM		Repeat last message.
QSO	–	Message for . . . .
QSP		Relay via . .
QTA	Cancel message to	Cancelled
QTC		The message is . . .
QTH	What is your address?	My address is . . . .
QTR	What is the correct time?	It is . . . UTC.
QTX		Log on.
QUA	Send me all new messages.	
QUC	Who did the last message i sent: go to?	It went to . . . ,
QBM	Send me the message from	
QDB		The message to . . . , is for warded
QJC	May I call . . . direct?	Yes
QJG		Revert to message mode (log off interactive mode).
QNO	–	Negative response or action.

Fig. 4. Basic NCL Dictionary

	Statement	Response (if any)
:QRA:	What is your call sign?	My call sign is ...
:QRG:	What is my exact frequency?	Your frequency is ... kHz
	What is the frequency of ...?	His frequency is . . . kHz
:QRH:	Does my frequency vary?	Yes
:QRK:		Your bit error rate is . . .
:QRL:		I am busy now, please call me later
:QRM:		Your signals were interfered with
:QRN:		There is noise on the frequency
:QRO:	Increase transmitter power	
:QRP:	Decrease transmitter power	
:QRQ:	Speed up to . . . bauds	OK
:QRS:	Slow down to ... bauds	OK
:QRT:		Signing off from the network
:QRU:	Have you any messages from me?	Yes, messages are from . . . .
:QRV:		Signing on to the network (includes QRU by implication)
:QRX:		I am busy now, I will call you later
:QRY:		It is your turn to send a message to ...
:QRZ:	Who is calling me?	
:QSA:		Your report is signal strength ...
:QSB:		Your signals are fading
:QSD:		Your signals were mutilated (negative acknowledgement or not received) try again
:QSG:		I cannot accept packets from ... stations concurrently
:QSK:	Can you operate full duplex?	I can operate full duplex
:QSL:		Acknowledging correct reception of packet
:QSM:		Repeat packet
:QSN:		I can copy you directly
:QSO:		The message follows
:QSP:		Please relay to ...
:QSU:		Send your reply via . *. (gateway or repeater station)
:QSW:	Which frequency/channel will you reply on?	I shall reply on ...
:QSX:	Can you copy . . . direct?	I can copy ... direct
:QSY:	Change to channel/frequency ...?	OK, I shall change to channel/frequency ...
:QSZ:		Repeat message or last packet
:QTA:		Cancel message
:QTB:		The character count is ... (used in RTTY messages only)
:QTC:	How many messages do you have for me?	I have . . . messages for you
:QTH:	What is your location?	My location is ...
:QTR:		Message was originated at (day, time)
:QUA:		Send me all the new messages
:QUC:	Has the last message I sent been forwarded?	Yes, it has been forwarded to ...
:QBM:		Send me the message from ...
:QDB:		The message to . . . has been forwarded
:QIC:		This is a direct call to ...
:QJG:		Reverting to automatic mode
:QNO:		Negative acknowledgement

NOTES: *:QSA: and :QRK: can form the basis for signal reports.*

:QSM: could be used to flag a message that has been passed via a store and forward repeater.

:QSU: can be used for routing control, whereas :QSP: defines final destination.

:QTA: is used by the operator to delete received messages from his system.

:QUA: can be used when transferring the function of network computer from one computer to another.

:QDB: could form an intermediate acknowledgement when tracking the routing of a message through the network.

:QIC: can be used to find out if a station is logged on at any particular time.

:QNO: is the standard negative acknowledgement to state that the receiving station cannot perform the desired operation; i.e., it cannot QSY or QRS.

This figure contains an initial proposal for the NCL dictionary which wfll, of course, be changed as NCL comes into use.

Fig. 5. Linking of Baudot and ASCII Networks

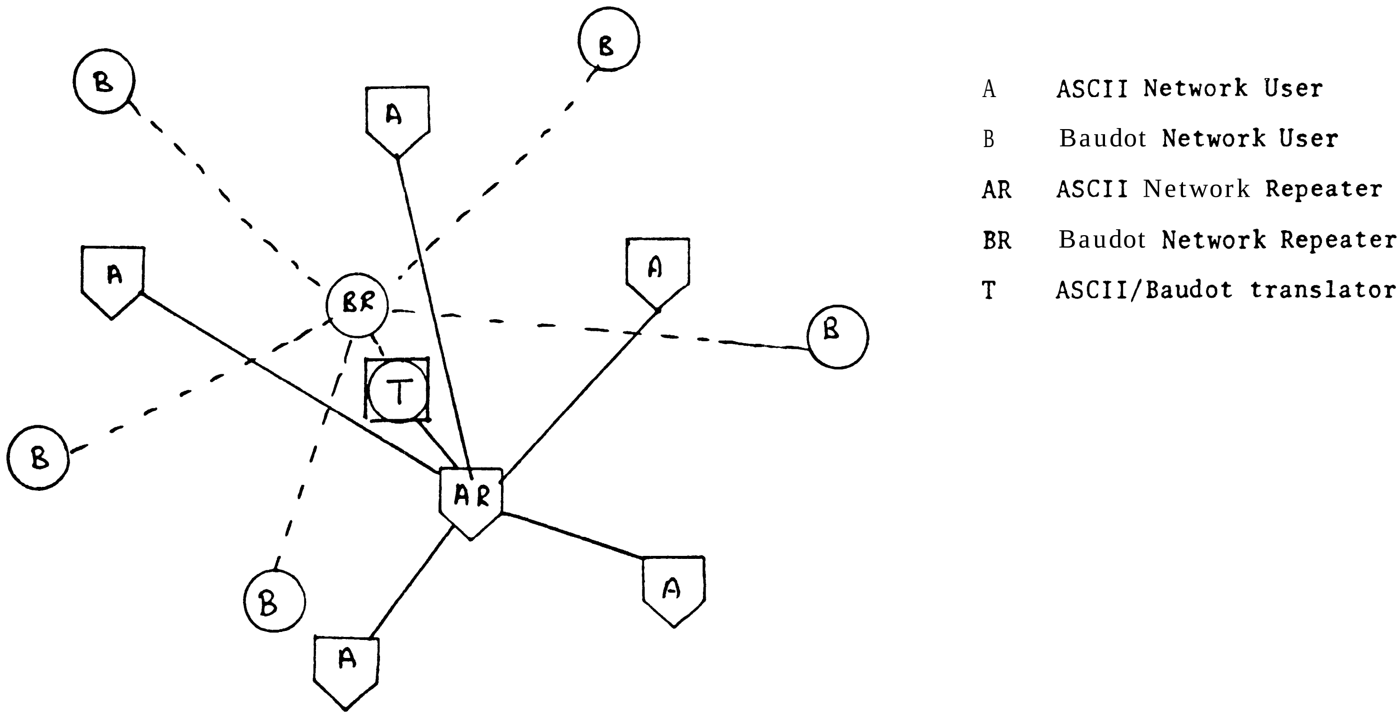


Fig. 6. A Basic Digital Repeater

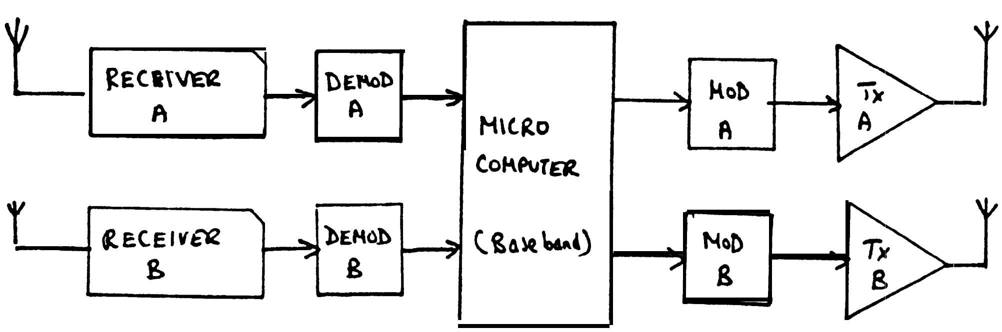


Fig. 7. Interlinking of Networks

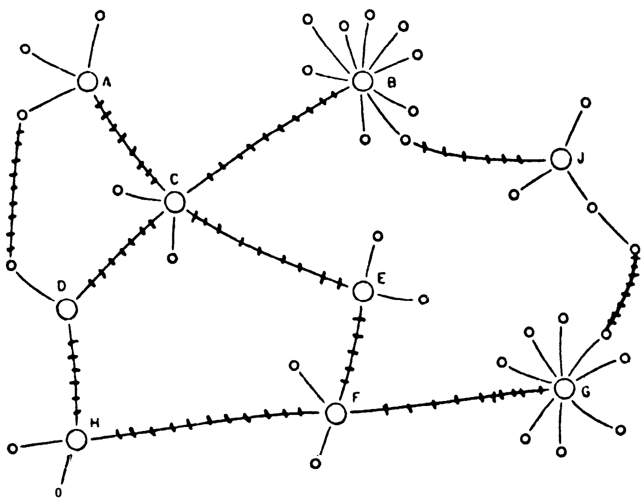


Fig. 8. Hf Propagat ion

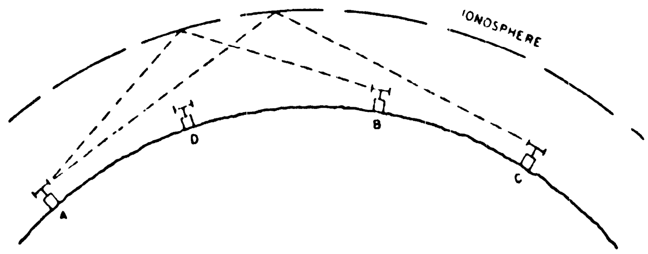
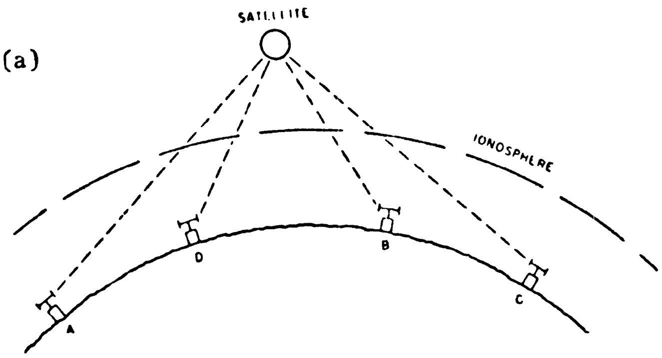
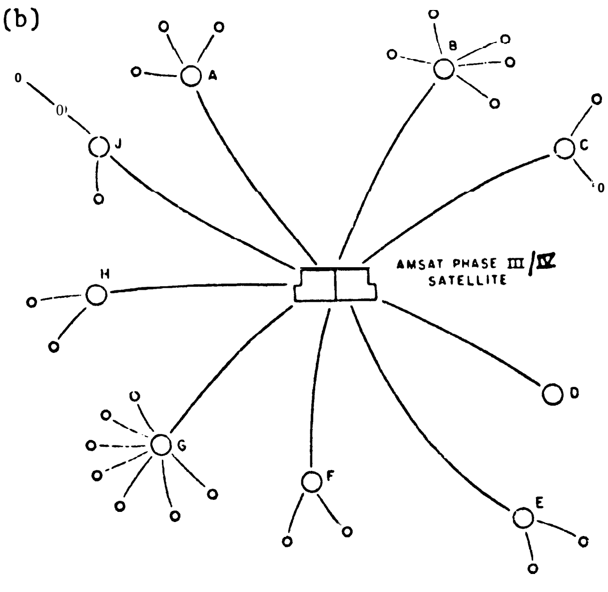


Fig. Fig. 9. .Satellite Based Network

(a)

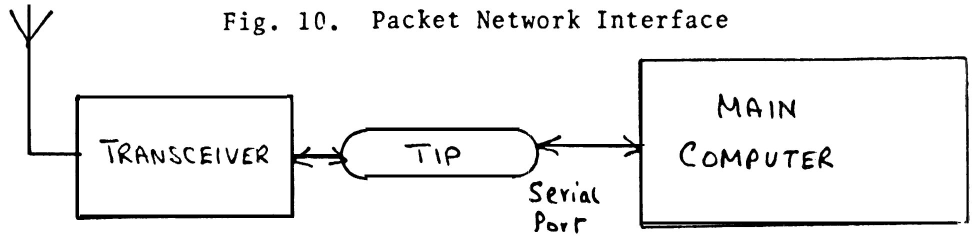


(b)



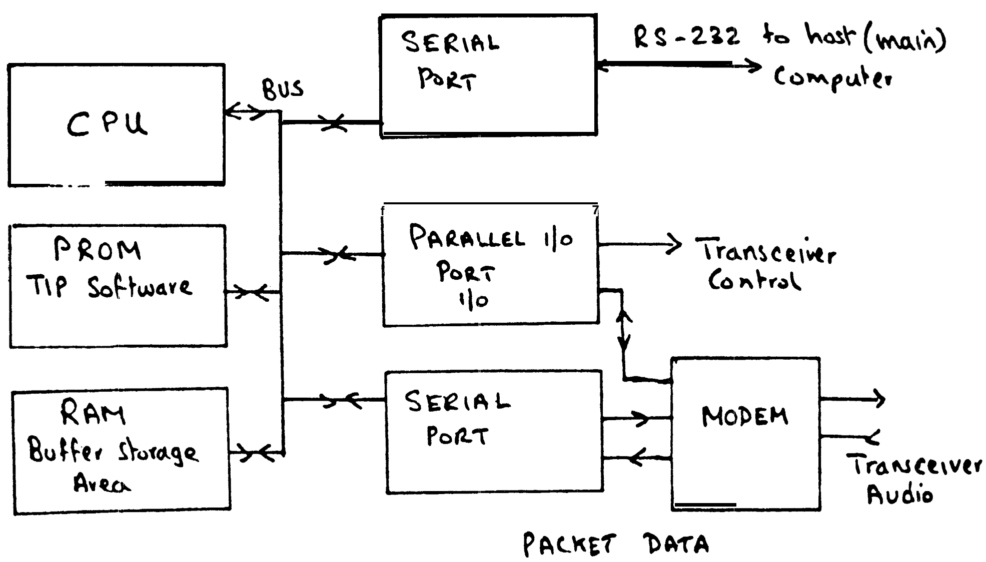
A distributed wide area network using a satellite. This approach would solve many of the message transmission problems. For a message to go from a station in network A to a station in network G it would hove to travel only through the satellite and hen be relayed to the other station.

Fig. 10. Packet Network Interface



Note: Similarity to RTTY TU.

Fig. 11. A Stand-alone TIP



Note: The block diagram is that of a typical micro processor module

NETWORK ARCHITECTURE AND PROTOCOLS FOR A WIDESPREAD AMATEUR DIGITAL COMMUNICATION NETWORK

Douglas Lockhart, VE7APU
29 Shamokin Drive
Toronto, ON M3A 3H7 Canada

In the last couple of years the Vancouver Amateur Digital Communications Group (VADCG) programmable communication controller has been used in many areas of the U.S. and Canada. As one of those who worked on the development of the board and its software, I am very pleased to see that it has gained fairly widespread acceptance in the Amateur Radio fraternity. It was not so clear, a couple of years ago, whether or not it would be accepted because it involved the use of techniques unused in Amateur Radio at the time. My impression of Amateur Radio then was that it had a great deal of inertia or resistance to change. But at the same time, like a massive body, once it gets moving it has a large momentum. Now I believe that Amateur Radio is moving into digital communications, and nothing is going to stop it. We need only to guide it to the best system that we can. And what is the best digital communications system for Amateur Radio? I don't think that anyone knows. The design of a commercial digital communication network costs hundreds of millions of dollars. That's right! - just for the design, not for implementation. Yet, even after all this expenditure, most commercial systems have their problems and detractors. So, in spite of the small amount of money that Amateur Radio will be spending on network design, we may still be able to come up with a system equal to, or surpassing, commercial designs. With this in mind I will outline the general philosophy of the system that we are working on in the VADCG.

Firstly, we wanted a low-cost interface to the network for an end user. We felt that a user should not need to have a computer just to access the network. For this reason, we designed, produced and programmed the VADCG programmable controller. Of course, there were many other good reasons for going this way, but I am mainly trying to show the function of the controller in the network.

The network that we designed the board for was not intended to be homogeneous but a network in which nodes would have different functions. Some of the node functions identified were:

1. A 'Terminal' or 'End user' node. Typically, someone with only a teletypewriter or video terminal, although a user accessing the network through a microcomputer would also qualify in this category (i.e. an intelligent terminal).

2. A 'Gateway' node. A node which allows users on the network to access another digital communications system. Examples:

A gateway to the telephone system using an auto-answer/auto-dial Bell 103-type modem.

A gateway to a digital communications channel on a satellite.

A gateway to the Local vhf RTTY channel.

A gateway to another amateur digital communications network.

Note that if a node is used to interconnect two networks which have the same protocols, it should not be called a 'gateway' because, in this case, the two networks are actually only parts of one larger network.

3. A 'Repeater' node. Used to extend the coverage of the network.

4. A 'Logging' node. To record activity on the network to satisfy regulations as well as for performance analysis.

5. A 'Host' node. This is the computer system attached to the network and is usually the system that the end user wants to use. It contains the programs and files that the user wants to use, such as editors, games, compilers, assemblers, file-transfer programs, files of swap-and-shop information, mailing list, etc.

6. A 'Station' node. Coordinates the operation of the other types of nodes in the network. Provides network services and communication between the network and the end user, repeater, logging, gateway and host nodes. At present, the concept calls for all messages to pass through the station node, but this is not an absolute requirement in order for the station node to do *its* job. The station node provides the higher levels of network protocol that the simple end user cannot provide for himself because of the limitations in storage capacity and complexity in the end-user interface.

7. A 'Message-switching' node. This is a node which has sufficient storage capacity to be able to store messages and data for an extended period of time. Such

a node would be something like a computerized bulletin board system (CBBS). Information which could not be directly transmitted to its destination immediately could be left here to be sent onward when the destination node was available. The communication network is a packet-switching network and so has little storage capacity for messages. Messages are sent through the network only when both the source and destination nodes are available. The message switching node could have messages to be received by any user on request as well as messages intended only for a specific user.

As you can see, the station node has a much more complex task than any of the other node types. Furthermore, the station node becomes almost indispensable in a system designed to use it. Because of heavy reliance on this node, it should be backed up by another station node in the area or by a repeater node allowing communication to another area which also has a station node. The hardware for this station node should be fairly reliable because it involves no moving parts. The station node being used by the VADCG, for example, is a three-card S-100 bus system. One card is a standard CPU card, another is a 64-k dynamic memory card -- both of these are standard S-100 cards which are readily available from many suppliers. The third card is a special I/O card which the VADCG has developed for handling the special needs of the station node. The card has four channels of HDLC communication using the Intel 8273 chip and six interval timers. The interval timers are used to handle line timeouts and to simulate a time-of-day clock. The timers and the HDLC channels are all interrupt driven using 16 channels of vectored interrupts provided by two AMD9519 chips. Also using the interrupt structure is the power failure circuitry, the transmitter fault-detection circuitry and the circuitry to detect software failures or loops. A failure in any of these areas will cause the CPU to enter a program contained in up to 8 k of EPROM storage on the same card. This program allows up-line reloading of the station node software or down-line dumping of the station node software for analysis of software errors. Each channel has a choice of baud rates and can operate with either synchronous or asynchronous modems. A number of extra control lines for input or output are provided to control external devices.

Some of the functions and services provided by the station node are:

1. Establishment and termination of virtual connections between nodes in the network.

2. Communication with the end user in plain language. For example, the station node will provide an explanation of why a virtual connection could not be made. It will interpret and act on network commands submitted through a terminal keyboard. It will provide a list of the status of other users signed onto the local domain or provide a list of users in another domain, for example.

3. Drive a logging node to record the connection/disconnection of the users of the network giving times and dates as well as usage statistics.

4. Drive a repeater node so that the repeater will do intelligent repeating of frames. Not all frames received by the repeater should be repeated.

5. Provide the higher levels of protocol required for an extended network for the minimum end user system.

6. Make routing decisions and keep, dynamically, information on delay times. The routing system as planned will use a distributed delta-routing system allowing multiple paths for communication between station nodes -- something like the ARPANET routing scheme. Routing decisions will be based on minimal delay time. Changes in delay times detected by a station node will be passed to adjacent station nodes. New station nodes in the network will be integrated dynamically and will be deleted when communication is lost.

7. Communicate with non-end-user nodes in the network using concise coded or formatted network commands suitable for computer generation and interpretation.

The above is not a complete list of functions provided by the station node. Others will probably be incorporated as the system develops, but this list should give the idea of what the function of the station node is.

It should be noted that the above six types of nodes are not the only types possible but only the ones which we have identified as being the most important at the present time. Most functions can be identified as belonging to one of these six types, even though there may be occasions where there is an overlapping of functions. See Fig. 1 which should help to clarify the relationship of the nodes. Each station node has a 'domain' associated with it. The domain is the set of nodes that the station is providing services for. The domain is typically a geographical area such as a city, but different station nodes may operate on different frequencies in the same geographical area. The lines between the nodes on Fig. 1 represent logical communication links at the data-link level of communication. Not all possible communication links are allowed. Direct communication is allowed only between a station node and another node type or between two station nodes. However, a repeater node may be used as an intermediate node in this communication. Any messages sent between non-station nodes have to be routed through the station node

in each domain. To some, this may appear as a harsh restriction on the communication possible, for, after all, there may be nodes in the domain that can communicate directly because of their proximity. To answer this let's look at the advantages of going through the station node and the reasons for communication with the station node.

1. Standardization of the radio link. Each node's equipment has to be set up to interface only with one point. This means that adjustments to the modem, power output of the transmitter, frequency adjustments of the transmitter, orientation of the antenna and various other requirements for establishing a communication link have to be set up for only one link. This avoids having a large amount of coordination with various other nodes. Once communication is established with the station node, no other concern for communication with the rest of the network is required.

2. Low power and directional antennas can be used for the link. No rotator is required, even if using a directional antenna.

3. Nodes which are out of broadcast range anyway would have to go through the station node.

4. Nodes which were using a different band would have to go through the station node to communicate.

5. Nodes which were using different speeds would have to go through the station node.

6. Nodes which required protocol translation would have to go through the station node. More on this later.

7. Nodes communicating outside of the local station node's domain would likely have to go through the station node.

8. All nodes using network services would have to communicate with the station node.

9. Establishment and termination of connections between nodes would have to be arranged through connection services in the station node.

The above considerations do not totally rule out the channel-sharing advantages of being able to have nodes communicating directly on the same channel as that of the station node when they are using the same speed and not requiring protocol translation and are within communication range. The protocol would have to be more complicated to allow these two types of communication to be carried out on the same channel and yet allow coordination by the station node. All I can say is that the present software does not coordinate communications on the channel which do not pass through the station node. However, the software does ignore all addresses which have not been assigned by the station node so that other digital communication can share the channel. It would probably be more appropriate that these nodes use another channel for their communication as they appear to have little need of the network.

You are probably wondering how the VADCG programmable communications controller fits into this network architecture because most users of the board are using software in the board which communicates directly from one end user to another end user. Well, in fact, this software which is commonly in use was written after the original software for the station node architecture had been written and was already in use. The terminal-to-terminal software is actually a modification of the original software to get it to work in a station-less environment, In fact, the hardware was limited to 4k of EPROM and 4k of RAM because the higher levels of protocol were going to be provided by the station node or by the host node. In spite of the general usage of the board for direct communication, it is still the intent of the VADCG to develop a network based on the station node concept. More circuits have been developed, and software has been written for use in this type of network recently.

As Fig. 1 shows, this architecture is distributed at the station node level but not at the lower levels. There are multiple communication paths between station nodes but only single paths between the station node and other nodes in a domain.

The VADCG board can be used in the terminal node, the repeater node, the logging node, the gateway node and the host node. However, it probably is not suitable for use in the station node due to limited memory and the fact that it is a single channel. With suitable programming, it could possibly be used as a type of front-end processor for the station node. The VADCG is developing separate hardware for the station node.

PROTOCOL LAYERS

THE PHYSICAL LAYER - This is the lowest level. It details the characteristics of the physical communications interface between the system components. We are adhering closely to RS-232-C standards in the use of connectors, pin assignments and voltage levels. But, in addition to the RS-232-C serial interface, we are providing a TTL-level parallel interface and a 20-mA current loop interface in order to accommodate the widest possible choice of end-user equipment.

THE DATA LINK LAYER - This layer manages the error-free transmission of frames over communication links between nodes in the

system. Most communication networks are using a system very close to the HDLC standard as is the system that we are using. This protocol is the same as that being used in the VADCG programmable controller for direct communication now. Unlike IBM's SNA, which supports only an unbalanced version of HDLC, we are using a balanced version in which neither node at each end of the link operates in slave mode. Both nodes share packet transmission and recovery responsibility. When this layer receives a frame in error according to the frame check sequence contained within each frame, it requests the retransmission of that frame and all following frames. The reception of each frame is acknowledged, and if no acknowldgement is received, some transmission fault is assumed to have occurred, and corrective action is taken. This is usually an additional request for acknowledgement. If additional requests for acknowledgment fail, then the link is assumed to have failed, and other corrective action is taken. The protocol requires positive acknowledgment only after every 7 packets. The establishment of the link uses an initial connection protocol (ICP) in which information is exchanged between the connecting node and the station node. The connecting node passes a description of itself to the station node which keeps it in a table for the duration of the connection. The station node passes an assigned data link address to the connecting node which is used by both the connecting node and the station node for the duration of the connection. The protocol is half-duplex, multipoint and uses a carrier-sense technique (CSMA) to resolve contention on the radio channel and improve throughput. The contention protocol used by the station node is slightly different than that of the other nodes in order to give the station node an advantage when contending for use of the channel. This is done because all traffic in the domain must pass through the station node. The station node is working for all the other nodes.

THE NETWORK LAYER - This layer provides services which transport data through the network to its destination node. Messages that are transferred between domains in the network require a full network address and network flow-control functions. This information is added to the beginning of the packet as another block of information creating what I call a type 2 packet. The packets coming from a simple end user do not have this additional information and are in a type 1 format. These services are provided in the station node but may be provided by a multi-user host node. The decision to support type 1 or type 2 packets by a host node is indicated at the time of initial connection. When type 2 packets are selected, no translation of packets is done by the station and the management of the destination source address fields as well as management of the sequence number is left to the host node.

See Figs. 2 and 3 for the layout of the packets.

The following is an explanation of Fig. 4:

After receiving a packet passed to it from the next lower level (the data-link level), the packet is translated into a type 2 packet using tables kept in the station node. The packet may already be type 2 in which case this translation is not necessary. The packet is then analyzed to see what its destination is. If the packet is not for this domain, then it is routed back to data-link control. The router uses routing tables kept by higher layers to decide what link the data should be forwarded on. If the packet is for this domain, then it is either for network services or for another node in this domain. If it is for another node in this domain it is translated to type 1 if necessary and passed to the data-link layer. If it is for Network Services, then it is checked to see that it originated from an end-user terminal. If it did, it means that the data has been typed in using English words and must be parsed and analyzed by Terminal Input Services before being passed to Network Services for action.

As a result of the commands received by Network Services, Network Services may have control messages of its own to send to various points in the network. These control messages use codes suitable for interpretation by a computer. If they are to be sent to another domain, then they are sent via the router to Data-link Control output. If they are for this domain and have to be interpreted by an end user (Terminal), they are passed to Terminal Output services which translates the codes to suitable English language sentences. The packet format is translated to type 1 if necessary before being passed directly to Data-link Control output. This technique has a couple of advantages. First, because a knowledge of the details of the characteristics of the terminal is kept in the domain's station node, Terminal output Services has all the information available to do fancy formatting of the message to the terminal. It knows the line length, whether the terminal supports lower case, highlighting, go to xy, erase screen, etc. This is not known at the remote Network Service point. Secondly, the computer format is more compact than the form put out by Terminal Output Services and so is more efficient at utilizing the longer communication channels.

Note that for every command that can be entered in through a keyboard by an end user, there is a corresponding coded command suitable for generation by a computer. Likewise, for every plain-language response to a command, there is a coded (or formatted) response for a computer program. This means, for example, that if there is a file-transfer program running in a host computer the file-

transfer program can establish a virtual connection with another node using the network commands, transfer data across the connection and terminate the connection without human intervention. Host nodes are capable of establishing multiple virtual connections at the same time.

DEVICE SUPPORT

As mentioned earlier, the station node receives and holds information on the configuration of each connected node. This information is passed to the station at initial connection. In the case of a terminal node, this information contains details of the device characteristics and addresses in the node. When a connection is established between an application program and a device, the application program can request the device characteristic information from the station node. On the basis of this information, the application program can decide how to communicate with this device or even if it is capable of communicating with it. For example, suppose a user tried to use a full-screen editor program but had only a hard-copy ASR-33 terminal. The application program can send an error message to the user and disconnect. On the other hand, suppose the full-screen editor program found that it was communicating with a video display, then it would need to know how many lines and columns were in the display, whether lower case was supported, whether highlighting was supported, etc. The full-screen editor would then be able to communicate with the video display efficiently. This exchange of information binds the device and the application program it successful. There will be commands available to the end user to dynamically change the device characteristic information after connecting to the station node.

SUMMARY

I was hoping to be able to go Into more detail on the routing, device support and packet formats in this paper but realize that each of these ought to be the subject of a separate paper.

The author feel that the station node concept of network development offers the most function for the least cost to the minimal end user. The specialization of function in the system prevents the waste incurred by duplicating the same code in every node. As new functions and services become available, they are instantly available to all users of the network. The routing decisions are made at the station node level, and the network is distributed at this level. This appears to be a reasonable tradeoff because the routing code is fairly complex and maintains a large amount of network information. Furthermore, there does not appear to be a simple distributed-routing system in the literature that is workable for the low-cost end user node. The many advantages that the station node offers appear to strongly outweigh the disadvantage of having to rely on it. In any case, we will have to rely on something if we are going to get our messages relayed across the continent reliably, and I am sure that Amateur Radio is going to have its own digital communications network operating across the continent before very long.

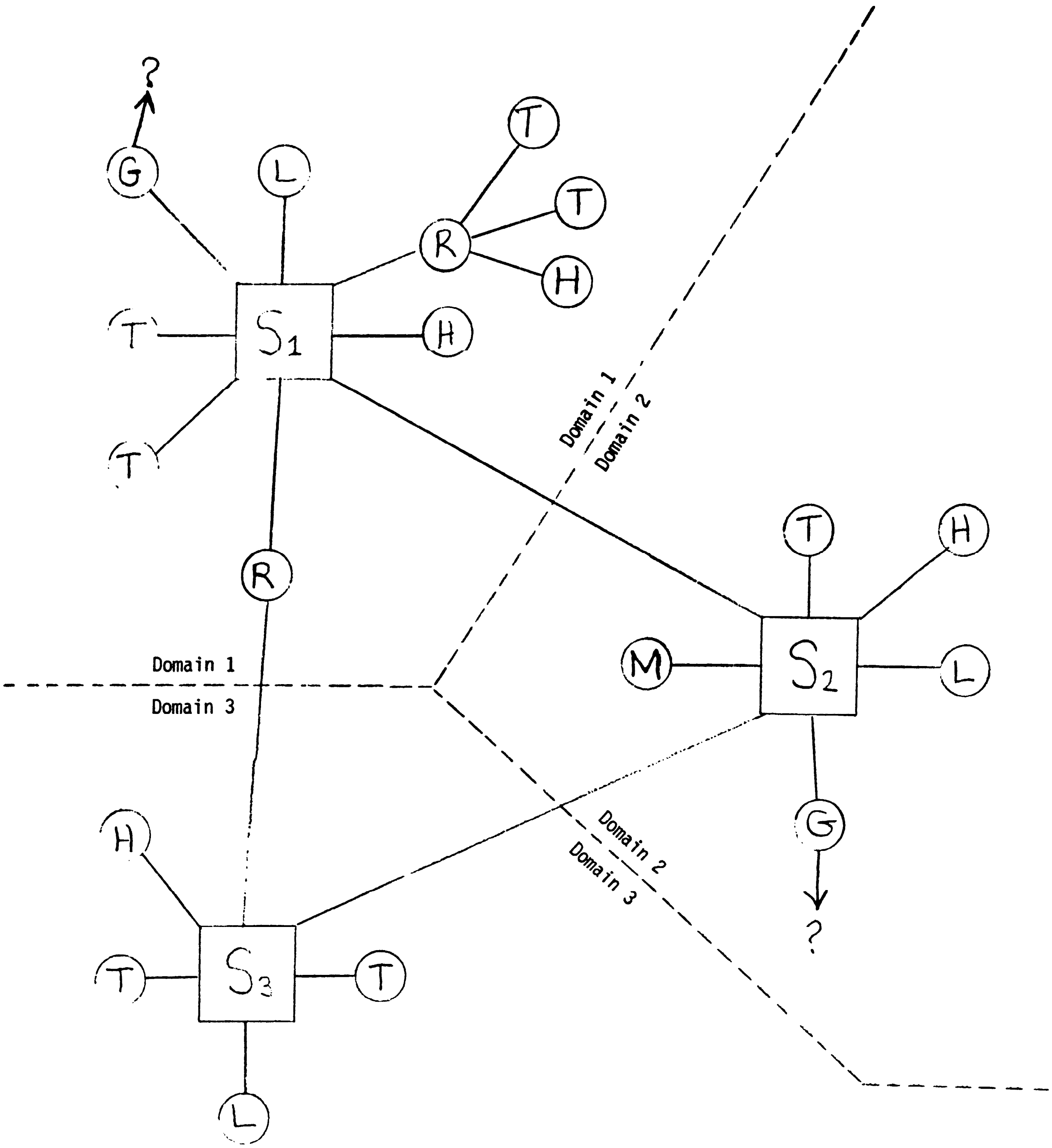
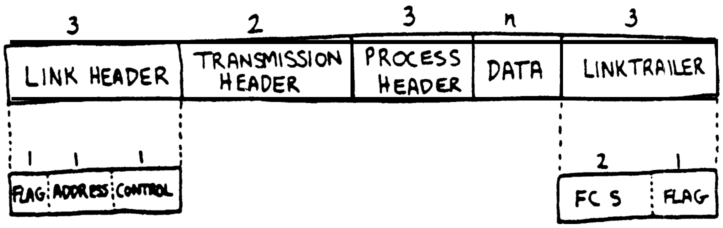


Figure 1. Relationship between nodes in a sample network.

H = Host, S = Station, T = Terminal, G = Gateway, L = Logging, R = Repeater
M = Message Switcher   ? = Unknown network or service

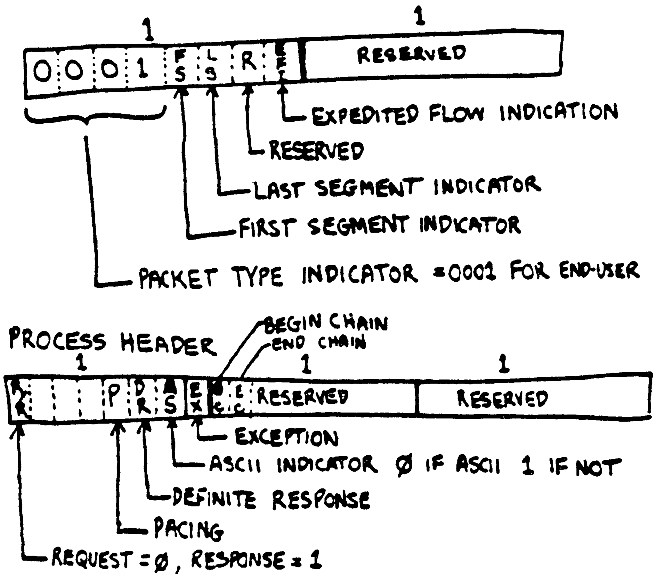
Fig. 2.

PACKET LAYOUT (END-USER TERMINAL NODE)



TRANSMISSION HEADER

Fig. 3.



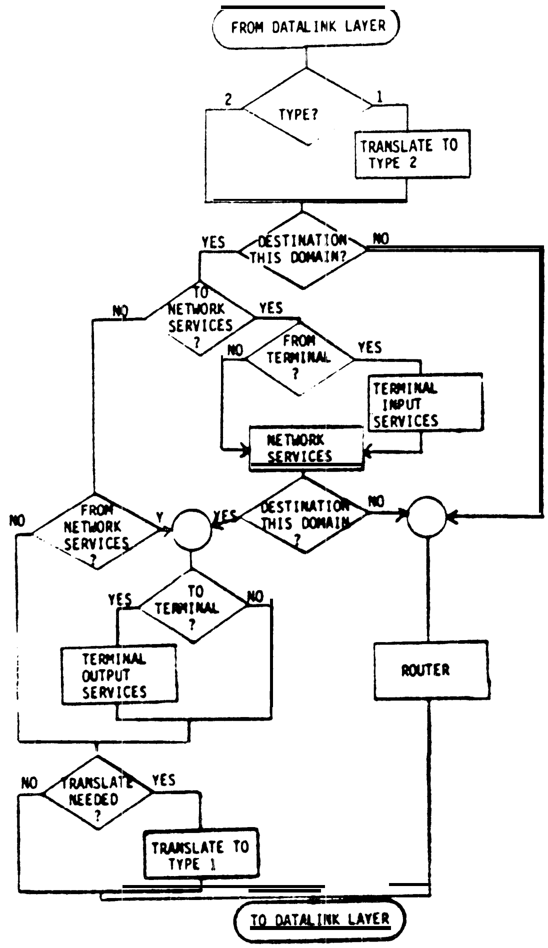


Figure 4. GENERAL DATA FLOW IN STATION NODE

ON THE CARE AND FEEDING OF YOUR PACKET REPEATER:

by

Rank Magnuski, KA6M
311 Stanford Avenue, Menlo Park, CA 94025
(415) 854-1927

This paper will describe the construction and functioning of the KA6M packet repeater, and will report on the operational results during the first 10 months of service. Since its initial turn-on in December of 1980, the repeater has been transformed from an experiment to a major Bay Area repeater serving a user group now approaching 30 stations. The repeater has been extremely important in testing new hardware and software, and in provoking interest in the area's amateur community.

I. The KA6M packet repeater

From Its initial turn-on in December, 1980, and through most of the Spring of 1981 the packet repeater was operating out of my residence in western Menlo Park, California, a location which is in the foothills which border the western shore of the San Francisco Ray. It was an experimental machine then, but could be heard well through most of the northern end of Silicon Valley, even though the power level was modest. The only station equipped to use it then was located in the same house, so there was never any real problem with signal path. Since then we have installed a couple of upgrades to the control software, we have used a better CPU card, increased the power level, moved the repeater up to a 700 ft. elevation, and Integrated its operation to be 100% compatible with the protocol used by the Vancouver Digital Communications Group's terminal node controller. The repeater has changed from being a laboratory curiosity to a major Ray Area repeater heard from Marin/Berkeley to south San Jose, and the user community has grown from a couple of stations to a network approaching 30 users. (See the April, 1981, QST for the initial announcement on-the machine.)

The KA6M/R repeater is currently operating on a simplex channel assigned (in the San Francisco Bay area) for non-voice use, 146.580 MHz., and transmits data at a speed of 1200 Baud. The machine consists of a Z-80 microprocessor, a Bell 202 compatible modem, and a solid-state transceiver with power amplifier.

The repeater hardware is based on STD bus cards. The STD bus uses 56-pin 4 1/2" x 6 1/2" card8 and is very popular for industrial process control applications. There are now many manufacturers supplying cards for this bus, and the repeater uses the Z-80/CPU-2 card from Mostek, which costs about $195. The STD card is very compact, and does not have unneeded extra circuitry which is typically found on more versatile personal system CPU cards. A WD1933 chip was breadboarded onto a Vector STD board, and one additional card to control the transmitter is all that was required.

A Bell 202 modem uses a mark tone of 1200 Ht., and a space tone of 2200 Hz. The modulation is AFSK-FM using unmodified voice-frequency rad lo equipment. The signal is NRZI encoded, which means that each zero data bit causes a mark-space or space-mark transition in the carrier. It is difficult to explain to old-time RTTYers that "mark" and "space" lose their meaning in an NRZI sys tern. The HDLC frame guarantees that there will be a sufficient number of zeroes sent so that bit timing and clocking can be recovered from the incoming signal. No start/stop bits are ever sent.

The radio is an PT202 handheld transceiver with a 20 Watt power amplifier. Since this repeater runs simplex, problems associated with full-duplex repeaters (desense, etc.) are not encountered, and exotic remedies, such as duplexers, dual antennas, and so forth, are not required.

The control software for the repeater is written in PASCAL/Z, a PASCAL which generates native Z-80 code instructions. Assembly language interface routines have been written to control a Western Digital 1933 HDLC chip. The software fits into two 2716 EPROMS, and 2K bytes of RAM memory are available for use. The repeater can store up to eight 128-byte packets.

Before giving any more details about the repeater, it would be worthwhile reviewing some concepts concerning packet repeaters.

XI. Packet repeaters in Amateur Radio Service

Virtually all repeaters in Amateur Radio Service can be classified into three major categories: Frequency-diversity, Time-divereity, or Space-diversity.

A frequency-diversity repeater is a conventional repeater which uses two or more frequencies for input and output.

A time-diversity repeater may use a single frequency for both input and output, but the signals are separated in time.

A space-diversity repeater or on-channel booster relies on the spatial. separation of its input ad output channels.

For digital data-communications work in Amateur Radio many variations on the above three approaches are possible!. In the frequency-diversity case , for example , we have

Linear translators - Devices which take incoming RF within a specified band and retransmit that RF on a new set of frequencies. There is a one-for-one mapping of each input frequency to each output frequency. Data communications through the soon-to-be- launched AMSAT spacecraft will use linear translators.

Audio transponders - The incoming RF is demodulated, and the recovered audio signal is used to modulate the outgoing carrier. Most 2-meter FM repeaters are of this type.

Digital transponders - The incoming RF is demodulated, and the resultant audio tones are converted to their logic '1' and logic "0" values. The digital signal is then used to modulate the outgoing carrier with the correct AFSK tones. Many RTTY repeaters use this technique; it discourages voice communi- cations. Digital signal regeneration may also be done at the character level, where a complete incoming character is received and then retransmitted with new tones and bit timing.

In all of the above examples, input and output activity occurs simultaneously, with practically no time lag in retransmission of the signal. Such repeaters are termed "duplex" machines, because the information is going in two ways at the same time.

In the time-diversity case there is local storage for packets or frames at the repeater site, and we can run the repeater in a "simplex" mode, **i.e.,** transmitting and receiving the frame on the same frequency. The outgoing signal is completely regenerated at the frame level, and all receiving stations are presented with a uniform signal, independent of the condition of the original input.

The primary function of a packet repeater, as with a more conventional repeater, is to extend the geographic range and coverage of fixed or mobile stations. A packet **repeater, however, can be programmed to** selectively repeat only those packets addressed to it, thus allowing the possibility of multiple repeaters on the same frequency, an advantage instead of a curse! In fact, there is a wide range of functions which can be programmed into a packet repeater, and we will discuss some of these options later in this **article.**

The FCC currently considers a packet repeater to be in the same category as an ordinary voice repeater) and so it must abide by the rules governing voice machines. Thus, it must be in the repeater subbands, must ID with a /R, cannot be used on a simplex HF frequency, etc.

Of the three or four packet repeaters currently in use in amateur radio service there are no two alike, and each of the different approaches has its own merits:

Contention Protocol Repeaters - Each station transmits on the channel without regard to the activity of other stations. If a packet collision occurs, the station will not receive an acknowledgement for the packet, will delay for a random period of time, and then transmit again.

Carrier Sense Multiple Access - Each station is able to sense the carrier of other users and waits until the channel is clear. Collisions will occur in this form of contention protocol, but less frequently than above .

Time Division Multiple Access - Each station is assigned a time slot for its transmission. Not used by amateurs yet.

Polling Protocol Repeaters - The repeater cycles through a list of active stations requesting traffic **from each one.** No collisions occur, but stations must respond to the polls in milliseconds and the poll list. must be kept short fqr good throughput.

The trade-off in these different approaches is channel utilization vs. centralized control and coordination. The tighter controls lead to better throughput, but the penalty is dependence on the functioning of a central station or on increased complexity of station equipment. **Since** the Amateur Radio **service** is a voluntary, non-professional operation, any scheme which depends on the 24-hour functioning of **a** single, central control station for all network users should be looked at very carefully. A philosophy which adopts looser controls and multiple or redundant repeat points will probably prove more workable in the long term.

The design of the local area network and repeater for each amateur community will probably be different and tailored to the needs of that community and subject to the technical preferences and biases of its designers. As long as the local net designers provide for future compatibility with whatever national networking scheme is adopted there should be no problem **with variations** of **a** protocol for a given region.

Here in the Bay Area **we** are considering the possibilities of a two-repeater system, one serving the north Bay, and one for the south. The alternative is a single **high-level repeater. We** need further study of our expected traffic patterns and of the interconnects required to nearby cities. The higher the repeater the greater the range and coverage, and consequently the increased traffic and throughput delays. Lower **level repeaters can handle** smaller groups of **users with better** throughput, but connects to distant locations may require double and **triple** packet hops.

Our design so far has stressed the ability of **users to be able to do direct point-to-point** connects to target stations if they are within radio range . **This can offload some of the traffic from the** repeater and allows **stations to** communicate with each other should **the repeater go down.** Note also that the **1200 Baud** throughput is *cut* in half when packets have to **be repeated, and is** down **to a third or less when two hops are** involved. Some of our next experiments will be with two repeaters in tandem so that we can **gain some** experience in coupling two machines,,

While we're on the subject it would be worthwhile mentioning some devices **which are very** similar to packet **repeaters** but which have different names:

Gateways - It is frequently necessary to convert a packet from one media **to another or to** transport a packet from one local net to another. A "local network" usually implies geographic proximity, a common. medium, and addressing which is unambiguous and at the lowest level of the protocol hierarchy. Each of the stations using a particular satellite channel might be a member of a different local net, and the satellite channel would be used to connect different local nets **around the globe.** A gateway would be used to convert a **satellite** packet to the format of another local net, **say,** the net consisting of all. the stations using the **KA6M** repeater.

Station Nodes - If a repeater participates in creating or dissolving a connection between two or more users, or if it has extensive information about the state of the network and provides routing directives to other repeaters, that device is normally called a "station node". The exact functions of a station node depend on the design of the network, but in general we think of it as a repeater which has been promoted into the first level of network management.

Host Node - There are many different types of services which might be offered on a packet network, the most frequently mentioned one being electronic mail. Services such as that should be done by host machines on the network and probably should not be incorporated into a repeater. The hardware of a repeater should be simple and reliable, and should not depend on disk storage or other failure prone equipment.

In summary, .. becomes clear that with a programmable device we can implement many different levels of complexity and service in a repeater.

III. Operational Aspects and Details of KA6M/R

The Z-80 was an obvious choice for the basis of constructing a microprocessor based repeater system. It is a very popular chip, and some interesting high-level language support is available for it. The STD cards are small and compact, and it is extremely easy to wire a new board for testing ideas,

The choice of the Western Digital 1933 chip needs more explanation. Quite a few manufacturers are now making HDLC/SDLC VLSI chips. Of the half-dozen or so available, only two have all the hardware required for easy use in an amateur radio environment. The special features required are NRZI encoding and an on-board digital phase-locked loop for timing recovery. The two chips which have this circuitry are the Intel 8273 and the Western Digital 1933. The Intel chip has a more elegant programming interface, but is slower and more expensive than the Western Digital unit. In quantity purchase the Intel part is about $35, and the Western Digital part is $25.

The Bell 202 modem operates at the maximum Baud rate permitted on VHF, and works very well with voice frequency equipment. Also, 202's are available from many different sources and can be found at surplus electronics houses.

The 2-meter band was selected because it permits 1200 Baud transmission and radio equipment for it is widely available. Keeping the entry threshold low for packet users was an important part in introducing the service here. It's quite enough to ask someone to buy a new controller board and a modem. Requiring special radio equipment on top of that would be a burden.

The repeater design assumed that the machine eventually would be on a mountain top, and that limited access to it would be the rule. Thus the design was kept simple, and minimum functionality was planned for the first installation. Virtually no manipulation is done on an incoming packet. The repeater allows arbitrary binary information to be present in the information field, and the only field which is manipulated is the address field. The repeater does not participate in the link-level protocol, and any acknowledgements or requests for retransmission must be done by the end-user stations. Initially the repeater was set up to repeat a single packet with up to 256 bytes in the information field. That was changed, however, and it will now repeat up to eight 128 byte packets. Storing and repeating multiple packets in a single transmission is important for compatibility with the current VADCG software, and is required when packets are hopping down a chain of repeaters. If two packets are coming from opposite directions, one of the repeaters at crossover point must be able to store more than one packet.

The address byte modifications which were adopted work as follows: the repeater will repeat packets which have an (address byte of hex 80 to 9F. In repeating the packet, the outgoing address will be the incoming address plus hex 20. Thus, repeated packets will have an address in the range A0 to BF. This is the only alteration to the packet, and is otherwise an exact duplicate of the incoming packet. The address change is required so that receiving stations are not confused by a packet coming both from the sending station and the repeater. Stations wishing to direct connect may use addresses in the range 01 through 7F. The VADCG TNC software has been modified so that a station receiving a connect message adjusts its outgoing address to match the repeater/non-repeater range of the incoming connect message. So, a station wishing to log into the mailbox can do so directly or be repeated, and the mail'box will automatically adjust itself to the selection made by the operator.

This addressing scheme has two major problems. First , it requires an individual to coordinate address assignments within a user community. Second , if the user population grows beyond the 32 slots currently assigned to the KA6M repeater, something has to give. Several suggestions have been made to let the repeater dynamically assign addresses, but what we're really facing here is the tip of the iceburg with respect to some real serious issues concerning local area and nationwide network addresses and how to handle them. Some bandaids could be applied to the addressing strategy described above, but I think it would be better to face the more global issues right away.

As soon as a packet is received the frame check sequence (FCS) field is checked, and if not correct, the packet is discarded. Next, the incoming address byte is checked for the right range, and if approved, the address field is modified and the packet is put on the outgoing list. When one or more packets are ready for retransmission and when the modem carrier detect signal has dropped to the channel clear state, the repeater turns on its transmitter and transmits all packets on the outbound list.

The repeater has circuitry to sample both RF Carrier Sense and Modem Carrier Detect. Originally packet retransmission waited for a clear RF channel, but this proved unworkable in an RF jungle such as Silicon Valley. Now, the beacon will hold back if RF Carrier Sense is on, but packet retransmission will only be delayed by other packet activity on channel, or by tones which can fool the modem, such as RTTY AFSK.

If at least 5 minutes have elapsed, and the

channel has been RF free for the last 30 seconds, then the repeater executes a beacon subroutine which transmits three 70 character packets. The beacon messages are very useful for testing the receive path of new packet stations, and for verifying that everything is in order both at a working station and at the repeater. They also have attracted lots of attention from hams in the community who have never before listened to packet transmissions. One uninformed gentleman described the beacon packets as "The longest squelch tail I ever heard".

Entering the beacon subroutine also triggers the CW identification. The hardware has separate control leads for request-to-send and transmitter on. The software currently turns the transmitter on and then toggles R-T-S on and off to produce the code required to ID. This procedure is not as good as the VADCG CWID routine, for packet stations can fire off a packet in the dead air time between the dits and dahs. The repeater, of course, is not listening. The VADCG TNC keeps R-T-S on continuously and uses Mark (1200 Hz.) for the dits and dahs, and Space (2200 Hz .) for the inactive time.

The transmitter control card has some simple push-to-talk control circuitry on it and 'a 30 second watchdog timer. An absolutely foolproof watchdog timer is difficult to construct because it would have a hard time discriminating between normal program operation and a loop which periodically turned RTS and the PTT circuit on and off. Now it only catches a continuous RTS on state. One really needs two or three independent checks on the program to insure proper functioning. Failure of any one of those checks should reset the repeater.

The FT202 radio has worked rather well considering its initial cost, and any similar unit would be suitable for small to medium size metropolitan areas. Crystal controlled radios, in general, are ideal for packet service. They tend to have faster switching times than the new synthesized units. We are going to have to upgrade the unit here, however, because of the dense RF in this area.

One of our more interesting findings is that the repeater has proven to be tremendously useful as a test tool for bringing up and checking out stations. The machine is there 24 hours a day, and by bouncing a packet off it you can quickly verify if your station is in order. The beacon messages provide a readily available signal source. The eight packets which can be stored represent nearly a third of a page of text, enough to trigger failures in T/O interfaces which might be prone to dropping characters.

We also discovered, and this sounds really incestuous, that you can self-connect through the repeater. Due to the way the VADCG and repeater software and addressing is structured a station can make a complete round-trip connection loop to its own equipment. A line of text typed in at a terminal will come back to the screen via the repeater, and all error checking, re transmit and flow control logic is active during such a test. This is an excellent way for a personal computer system to exercise its software and test to see if there are any conditions under which bits and pieces might be dropped.

The repeater users have found that packet radio service is more demanding of their stations than originally expected. Modem programs that used to work fine at 45 or 300 Baud may not be able to keep up at the 1200-9600 Baud rate coming from the TNC. File transfers are expected to be 100% reliable, so the flow control mechanisms have to function perfectly in order to not lose data, Radios, modems and antennas have to be adjusted for optimum settings if the 1200 Baud 202 modem rate is going to be meaningful. We have reached the point where most of the pieces of our network are working, but a lot of fine tuning remains to be done.

IV. Advanced Funstions for Repeaters

The following are some of the ideas which have accumulated but which are not currently implemented in our repeater:

Program bootstrap - A ground station is used to load the bulk of the repeater program. New features and functions, program corrections, routing tables, etc. could be maintained at a convenient host station away from the repeater site.

Changeable beacon messages - Special packets could be sent to load new messages of importance to the repeater users.

Time of Day - A realtime clock. could transmit current time and date or GMT.

Statistics - The repeater could keep track of the number of packets sent, packets in the last hour, number of users, total traffic, and so on.

Diagnostics - The repeater could be put into a mode where it would send nearly continuous packets of some interesting pattern for a period of time. This would be useful for checking modems and levels and continuous reception of data at: a station,.

Log list - Users like to know what stations might be monitoring and what stations were act ive in the recent past, The callsigns of the last ten connecting stations could be kept in a first-in first-out list,

Multi-party conferences - The current point-to-point connections provided by the HDLC link level protocol do not allow real--time conferences or round tables. The issue, of course, is data integrity. Error checking, and retransmission must be done for each user plugged into the discussion. A smart repeater or station node might be able to arrange such a multi-way conference.

Terminal-node repeaters - Any user who desired or was in a strategic location could program his TNC to act as a repeater in addition to the usual TNC functions,

Role in larger network - The real mission of a repeater is to be a member of a larger national or international network. What functions have to be added to the program logic of a repeater to accomplish this is a subject for debate, and will hopefully be outlined in the near future.

V. Future Directions

There are many possibilities for future development of repeaters in the amateur service:

Hardware - High speed, 48 Kilobaud UHF repeaters using direct digital modulation of the the RF, with no audio modems used at all. Decoding will be done directly from the IF strip. Repeaters with dual RF modules and selective control of frequencies and antennas. Spread spectrum transmission, with selective targeting of a repeater by changing the transmit hop codes.

Software - Development of higher leve 1 networking protocols in higher level languages such as PASCAL, C, ADA, FORTH, and others. Interconnects with other networks and CBBS's. Development of broadcast protocols which handle real-time conferencing and networking games.

It seems to me that we are only beginning, and that there is still plenty of room for future experimentation.

VI. Acknowledgement

I wish to thank the amateur radio operators of the Northern California area for their contributions to this report, and thanks also to Woody Teague, WA6TAT, for the current repeater site.

The software and schematics will be available from me for a modest charge. Contact me if you are seriously interested in duplicating this equipment.

On The Use of a Two Frequency Traditional Voice Repeater for Local Area Packet Networking

David W. Borden, K3MMO
Route 2 Box 233B
Sterling, Virginia 22170
(703)450-5284

Abstract

In using the VADCG Terminal Node Controller (TNC) board in the Washington D.C. Metro area, members of the Amateur Radio Research and Development Corporation (AMRAD) have found it convenient to use an existing voice two meter repeater for packet work. This paper addresses some of the problems encountered with this approach and some of the benefits that accrue.

Introduction

In traditional two meter amateur radio voice communication, a two frequency repeater is employed to facilitate communication between stations that are not in line of sight distance of each other. The output of a receiver tuned to one frequency is fed into a transmitter tuned to another frequency. A carrier operated relay on the receiver keys the repeater transmitter. A duplexer is used to allow a single antenna to be used for both receiving and transmitting. Some delay is usually included on access and the transmitter remains on for some short time after loss of input. The two frequencies are seperated by 600 Khz on two meters by agreement. AMRAD's voice repeater operates on a frequency of 147.81 Mhz transmit and 147.21 Mhz receive. For several years it has been used to share voice and data, often successfully.

When members of AMRAD began packet radio work, it was a natural evolution to use our existing voice repeater to experiment on. A number of problems surfaced that were not readily apparent.

Problems Encountered

AMRAD packet radio activities began with Bill Moran (W4MIB) purchasing a VADCG TNC board and convincing several others to join him in packet experimentation. The actual building of the boards went rather quickly once the two critical parts (the 3273 Protocol Controller and 8250 USART) were procurred. Surplus modems were obtained by our local club modem buyer who searched each hamfest at the crack of dawn to obtain Bell 202 devices for us to use on the repeater. After several false starts, the modems were correctly hooked to the TNC boards. Transmitter keying was a problem if the surplus modem selected did not handle Request-to-send and clear-to-send signals correctly. The speaker audio output of our two meter rigs often was not flat across the range of 202 modem frequencies (1200 Hz to 2200 Hz) and one tone or the other is attenuated. Transmitters do not transmit at once when commanded. If a relay is involved, many milliseconds could elapse before RF appears at the antenna jack. These small problems were surmounted and W4MIB transmitted the first packet on our repeater.

As soon as the second user tried to receive W4MIB's packet, the next problem was evident. The repeater was not coming up quickly enough. The machine would not come up fast enough to catch the HDLC preframe sync or flag. It caught the data and frame check sequence as well as the closing flag and CW-ID, but nothing appeared on the receivers terminal screen. Hank Magnuski (KA6M) supplied a fix to allow RTS to go high and turn on the transmitter, then delay a while until the repeater got the idea to turn on and repeat. This simple fix allowed packet work to begin in earnest and new users appeared. Quickly the demand for a QST packet every 8 minutes was evident. New users had to have something to tune up on without waiting for the nightly packet sessions.

Doug Lockhart (VE7APU) had supplied a program to do the QST packet and it was quickly implemented at the QTH of Terry Fox (WB4JFI). Another problem quickly surfaced. We received the QST correctly most of the time. However, when we left the shack and came back some time later, the screen was filled with last lines only. The QST packet had six lines! Where did the other five lines go? Sandy (WB5MMB) correctly laid the cause to the repeater ID. The repeater ID fires every 10 minutes when people are using the machine. But if no one uses the machine for five minutes or so, the repeater ID will not fire until the first access. The long dry spells of no activity were broken every 8 minutes by the QST packet which fired the repeater ID which squashed the first five lines of the packet. Sandy fixed this problem for us by decreasing the volume of the repeater ID tone and lowering the audio frequency of the tone. Now all six lines of the repeater made it every time......unless.....noise appeared on the repeater input at the same time as the QST packet. The QST packet is now a good indication of the circuit condition. If all

of us are consistently receiving all six lines of the QST packet, conditions are right for super terminal to terminal packet operations.

Voice interference is a problem in packet work on a shared repeater. This comes in two forms, intentional and accidental. Accidental interference occurs when two voice users think the packets (usually very short) ar noise bursts and transmit away on top of the packets. They win in FM voice operation if they capture the repeater input with a strong signal. One night when a voice user appeared in the middle of a packet QSO between W4MIB and myself, we carried on for 20 minutes by interleaving our packets in between their voice transmissions. We would wait until one voice user dropped his carrier and before the repeater dropped we would bang the line feed key to fire off the packet. Finally the voice users caught on. I think we got away with it for so long because some repeaters transmit a small tone when a user drops his transmitter. This tone allows the receiver to know it is OK to transmit. We carefully explained packet radio to the voice users who have never been seen again on the machine.

Intentional voice interference (not really malicious) occurs when voice users just cannot wait until the packet QSO is over. They sneak in a quick call to their buddy between packets (same trick we used in reverse in the previous example). Malicious voice interference occurs when someone fires the autopatch (the great bugaboo of data people) in the middle of a packet session. Channel sharing between packet and voice could work if users would let it.

Design Problems

Two frequency repeaters must be coordinated with other users in the area. This is done through local repeater councils who allot frequency pairs. There usually are none to allot on two meters. We already had ours and thus did not need a new allocation.

Startup expense for two frequency repeaters is high. The duplexer is typically $500. Problems get worse from there, ask Sandy, one of our control operators.

Advantages

Single frequency packet repeaters of the type constructed by Hank (KA6M) suffer from a hidden transmitter problem. In packet work, the modem tones are sensed (carrier sense) and other packet users do not transmit if they sense a modem is on frequency at the time they wish to transmit. If a given station cannot hear all other people on the repeater frequency, collisions are certain. A two frequency machine does not have that problem as all users hear each other (at least their modems do, the software ignores direct packets when connected).

AMRAD users find it convenient to develop software for the TNC using the AMRAD two frequency already available repeater. We are planning a single frequency repeater, but the software can be checked out before emplacing it at the new site.

The greatest advantage to using an already owned voice repeater is startup costs are already depreciated probably. It is inexpensive if your already have it - serendipity.

Possible Enhancements

If your group can afford the luxury of a two frequency packet repeater, two enhancements can be added at once.

First, never allow any voice. Use voice on your normal repeater to service the data channel if required. Accept no substitutes on the input. Only 1200 Hz and 2200 Hz tones, nothing else.

Second, allow multiple inputs on the repeater (phone line, HF, etc.) and transmit all inputs on the output.

Other ideas need to be tried. All input modem tones of the correct frequency should be repeated, thus no "request for repeat" bit is required and hard addresses coded up to 127 users. A logging computer should measure thruput and report to any user submitting a query.

Conclusion

It has been the experience of AMRAD that a shared voice and packet repeater attracts new packet users who want to see what is being sent. in those packets. One new user is worth a few intentional voice jammers. The newcomers ask questions and some eventually get up in packet. The QST packet every eight minutes, in addition to acting as a circuit tester and new user board checker, attracts questions. It is easy to find a retired ham to watch the device at his home and insure it is well fed. Most people who use the AMRAD machine own or use some kind of computer. These are the type of people we need to attract to packet data communications.

Formation of local standards in North Jersey Packet group

Stephen E. Robinson, W2FPY
47 Serpentine Rd.
Ringwood, NJ 07456
201-835-1152

The successful introduction of digital packet communications to amateur radio depends not only upon the technical standards of the hardware and software of the packet interface, but also upon the resources and needs of the local amateur community. The planning of a local packet network requires examination of factors such as the available engineering talent, financial resources of both individuals and the packet group as a whole, and pre-existing user equipment.

The introduction of new modes of communication to amateur radio is not without its history of birth-pains and trauma. The most common difficulty is redundancy with pre-existing modes. Single side band was in direct competition with AM for HF activity. Similarly, FM operation was at first rejected by those using AM on thr VHF bands. Today, NBVM lacks broad support because *it* has only marginal advantage over the existing, successful ... . Highly specialized modes such as ATV have no counterpart, and are assured continuity despite their cost of entry. Amateur ratellite activity, EME, meteor scatter, auroral propagation, and microwave operations are currently pursued by most amateurs on the basis of the challenge they pose rather than their value as a means of communication. Extrapolating this comparison, packet radio communication augments, but is nonetheless in direct competition with RTTY as well as with the phone and CW traffic networks. In this approach to the formation of a local packet network, we will take advantage of the user support of these modes rather than attempt to compete with them directly.

Packet radio could probably survive at its present level of activity without participation of non-technical users. If this is to be the case, it is unlikely that we will see the introduction of high-quality commercial packet radio transceivers. On the other hand, by making it possible for the largest number of users to get started in digital communications, it would be possible to support the cost of sophisticated host equipment in much the same way as FM repeaters are currently supported by their members.

The development of local packet network standards presented here is founded upon the needs of the non-technical user rather than those of the avid experimenter. This network is designed from the bottom-up (end user) rather than from the top-down (state-of-the-art technology).

We will start this design cycle by examining the minimum equipment necessary for digital radio communications. These are: a modem, a CRT or printing terminal (preferably ASCII), and a radio transceiver. According to the recent ARRL-sponsored survey (4), approximately 11 percent of radio amateurs are currently active in personal computing, and 4 percent are active in RTTY. Those with personal computers either have or can easily emulate a terminal. The RTTY users already have terminals, but require special consideration because of their use of Baudot rather than ASCII encoding. Note that I have deliberately left out the requirement for a protocol controller because some of the features of packet communications (storage and forwarding of messages) can be implemented on the host system on a single-user-at-a--time basis.

In an attempt to include non-technical users as well as experimenters in local packet radio activity, we have chosen a multi-layered approach to the network. The entry level to packet radio must be as simple and inexpensive as possible, and should use pre-existing equipment. The lowest level must use the host machine as the protocol controller. we call this entry level the "DUMBNET" because it is friendly to users having a “dumb terminal" and a 300 baud modem capable of half-duplex operation. DUMBNET will utilize asynchronous transmission format with no error checking, and will be similar to the now popular computer bulletin board (CBB). Several ports will be available into DUMBNET; these are: dial-up and one or more VHF or UHF FM repeater channels. In addition, by interfacing one of the host’s radio frequency ports to accept Baudot code and 170 Hz shift, those amateurs with existing RTTY equipment could access the net.

With the host acting as controller, traffic originating from these ports could be passed to other DUMBNET users, or to the more sophisticated true packet network users who operate off the same host, but on different frequencies. Message traffic on the net could be conveyed down to all DUMBNET users by means of a periodic role call. This role call, running perhaps once every 5 minutes, would list the amateur call signs of all stations having pending messages. Retrieval of messages from the host would be activated merely by sending one's call letters. In this way, those amateurs with dumb terminals will have an efficient way of visually “filtering” the

presence of messages bearing their station as the destination.

The next level up on the network, still using the same host as DUMBNET, is "SLOWNET". Using true packet protocols, ALOHA fashion, SLOWNET will interface more sophisticated users at minimal cost. SLOWNET will appeal primarily to those amateurs who have personal computers and can write or obtain software necessary to implement a simple protocol. Minimum equipment must also include, as in DUMBNET, a 300 baud modem capable of half-duplex operation. In addition, the user must provide a computer interface for controlling the transmit/receive function of his radio equipment. This does not represent major surgery to most personal computers or to the radio equipment. The modem transmit tones and T/R switching functions can be connected to the FM transceiver via the microphone cable, and the received tones can be obtained from an auxiliary speaker (or headphone) jack; in most cases, there will be no "cosmetic" changes to the user's transceiver. The packets will be in asynchronous ASCII format with an appended error-detection code.

Functionally, SLOWNET will not provide rapid communication for even a small number of simultaneous users. It will however provide excellent service for unattended message handling. The ability to send and receive personal messages and club bulletins will be the most attractive feature of this level of operation.

Still using the common host computer, but operating at much higher speeds, will be FASTNET. This portion of the network has not been worked out in detail, but will probably conform more closely with standards developed elsewhere (1,2,3). FASTNET will use synchronous ASCII format with error-detection codes. Because of its speed, 1200 baud or higher, FASTNET will be a challenge to amateur technology and ingenuity insofar as design of the radio interface and modem. It is likely that existing commercial high-level protocol controller boards will be used for the digital portion of the user interface. It is hoped that FASTNET users will be able to dedicate their equipment full-time to the net. If this is possible, then perhaps each user could eventually serve as a node in the net (all this transparent to the user).

The design goal of FASTNET is to provide rapid switching of a large volume of traffic over a limited coverage area. At this level, "packet-ragchewing" could be a reality.

Highest on the network scale is the interface for packet messages entering and leaving the area of local coverage. On this level, compatibility with other packet systems will be essential. This interface (OUTNET?) may connect to amateur satellite transponders, HF (low-speed) packet nets, or to similar host machines on VHF through microwave, thus extending coverage to other areas. The choice of protocols, baud rate, frequencies, and modulation schemes will depend upon the de-facto standards that will arise from today's experimentation.

In conclusion, this multi-layered approach was chosen not only because it reaches the greatest number of users, but also because it contains a chronological sequence for painlessly bootstrapping up an operational packet network. The multi-port host described here does not yet exist. On-the-air tests will be necessary to find out if a small (8-bit) microcomputer will be adequate for real-time response to all inputs, or if a larger computer or a distributed processing system will eventually be necessary. Since the host computer will define the character of the local packet radio network, the greatest development efforts will be placed here.

I do not present this approach for use as a "standard" for local networking, but rather as an encouragement to individual approaches that best suit the local user community. Of course, standards will be necessary for packet communication outside the local region, but to prematurely set standards for thre individual user may discourage experimentation and is contrary to the amateur spirit.

Acknowledgements:

The author is indebted to the participation of Gordon Beattie, WB2CAM, and Mike Friedman, WB2WNX, in planning this multi-level network, and to Phil Karn, KA9Q, for his many helpful discussions.

References:

1. Bordon and Rinaldo "The Making of an Amateur Packet Radio Network" QST, Oct. 1981, p. 28

2. Hodgson "An Introduction to Packet Radio" Ham Radio, June 1979, p. 64

3. Rouleau "The Packet Radio Revolution" 73, Dec. 1978, p. 192

4. Sumner "Survey of Amateur Radio, 1980. QST, March 1981, p. 11

An Emergency and Routine Communication Network for Illinois Using Packet Radio Techniques

Richard W. Doering, WA6CPM
1037 Cornish Drive
San Diego, CA 92107

Forrest R. George, W9SKD
16219 S. George Court
Plainfield, IL 60544

ABSTRACT

Amateur Radio teletypewriter and computer communication may be enhanced using a packet radio network to time share network nodes, detect and correct errors, expand geographic coverage, and facilitate bulletin broadcasting. The network is based on a common channel carrier sensed multiple access 0 persistent protocol. Application of such an area packet radio network to emergency and routine communication in northern Illinois is discussed.

INTRODUCTION

An excellent opportunity to enhance amateur radio communications for emergency and routine communication is now available to amateurs wishing to experiment with packet radio networking. Imagine, for example, approximately a dozen amateurs located within about one or two hundred miles of Chicago being able to carry on a half-dozen RTTY conversations, seemingly simultaneously, on a VHF frequency. If a heavy storm or tornado approaches one amateur, information concerning the storm could be automatically relayed to the other amateurs and the National Weather Service. General interest QST's, such as the ARRL bulletins. could be broadcast to the group. All this communication between amateurs who have either Baudot or ASCII RTTY systems running at almost any speed with error checking and correction! Using an inter-city land or satellite network, these dozen hams could even communicate with other amateurs around the country or the world.

This paper is intended to describe a proposed local or regional packet radio network, with special emphasis on the advantages of applying packet radio techniques to RTTY communications in Northern Illinois for emergency and routine communication. It is hoped that this article, and the others referenced, will tantalize the reader with the advantages and opportunities in experimenting and communicating using a packet radio network.

PACKET RADIO - A REVOLUTIONARY COMMUNICATION NETWORK

A Brief Tutorial Overview

Amateur radio communication has several forms:

- Casual Conversation
- Point-to-Point Messages
- General Amateur Bulletins
- Emergency Messages or Warning Broadcasts
- Experimentation

Each message has a priority, an orginator, and destination station or stations. Messages are generally standarized in format, for example, the messages in the National Traffic System. A protocol of message handeling proceedures is used to move these messages through NTS.

Protocol

Similarily, in this context of automated RTTY communications and repeating (packet radio networking), a standarized format is used for the messages. Messages are broken into packets of' a convenient length, each having a standarized format (ref. 1). Of most interest in this generalized packet is the following header and trailer information:

- Address - destination station (with node or zip code) sending station
- Priority - indicating message urgency
- Sequence number of Packet - used for message reconstruction
- Checksum - eg. Cyclic Redundancy Check for error detection

A complete discussion of protocols and packet switching is beyond the scope of this paper. One reference (ref. 2), besides others referenced elsewhere in this paper, describes protocol layers and packet switching in a readable way. Like structured computer programming of subroutines, layering protocols allows increasing complexity to be implemented while keeping order and transparent operation.

The protocol (ref. 3) currently being used by the KA6M "repeater" and the Vancouver Digital Communications Group (VADCG) seems to lack provision for priori-ty message designation and handeling. Recognition of *emergency* traffic would cause stations to temporarily reduce or suspend non-emergency traffic so that the channel would not be overloaded.

Network

Packets are transmitted from station to station through a network of nodes:

- Terminal node - individual stations
- Repeater node - simultaneous repeating node (separate I/O freq)
- Store-and-forward node - (memory at node stores message for retransmission on the same freq.)

The network and station protocol determining the permissible actions or transmissions of the nodes will be crucial to successful network operation (see ref.4).

Interfaces for Network

Since each message may be broken into several packets, each with sync, header, data, and trailer information, to be sent at high speed according to the network protocol, a computer-based interface is needed for transmission and reception. While some amateur radio operators who already have a computer may sacrifice their time and computer for dedicated programming on a packet network, most amateurs will find a stand-alone interface board desirable for their terminal node packet radio use. This interface board, known as a terminal node controller (TNC), will contain a microprocessor with some RAM, and a multiprotocol serial communications controller (a very smart UART). The TNC (ref. 5, 6, and 7) has EPROM-resident software to convert the terminal data (almost any speed, any code, ie. ASCII or Baudot) into the serial data following the network protocol.

The radio used in a packet radio network should ideally have a short turn-around delay (less than 10 msec,) to reduce dead time between packets and reduce collisions (see ref. 4 for a detailed description of the radio requirements), Commercially available radios will require modifications to achieve low turn-around delay. Typical radios may be used, however, the 400 msec. turn-around time will reduce message throughput.

Repeater or Store-and-Foruard Nodes

A packet radio network will generally rely on multiple "repeaters" on the same frequency to relay packets, using the store and forward method- A store-and-forward node eliminates the need for costly duplexers. By receiving packets, and checking the packet for agreement with the checksum, error detection may he performed. Errors are corrected (by all nodes) by the lack of an ACK acknowledgement for a packet causing automatic resending, (In packet radio systems, where there is multiple contention for the channel, the transmission of a NAK for negative acknowledgement only increases collisions with subsequent data.)

If only one "repeater" is to be used in a particular packet radio network, then the conventional simultaneous retransmission repeater may be advantageous. Such a conventional repeater may receive greater support by certain groups due to the opportunity tc repeat voice in additiot to packets, in some cases simultaneously. In this case the terminal nodes would ACK each other's packet through the repeater. A conventional repeater does limit the network in that multiple repeating, multiple routing is not easily supported.

Hank Magnuski, KA6M has succinctly stated the attributes of digital repeaters (ref. 6):

"The repeater serves to increase geographic range due to its advantageous location, it digitally regenerates the packet, providing all stations with a uniform signal, it selectively repeats only those packets addressed to it, allowing the possibility *of* multiple repeaters on the same frequency (an advantage instead of a curse!), and its beacon and packet-repeat facilities allow stations to do full-loopback testing, an invaluable resource in bringing up new equipment and checking out hardware/software modifications."

Features of a Packet Network

A packet radio network, like other forms of cooperation in our society, has much more utility as a system than the sum of its individual parts. Several important features of a packet radio netowrk over an ordinary serial communication link include:

- TDMA - Time Division Multiple Access
- Multiple Retransmission & Multiple Routing
- Error Checking
- Global or Selectjive Broadcasting

TDMA & CSMA

An attractive feature of a packet radio network is the nearly simultaneous use of the network by several commuaicators. The transmission rate through the network is chosen to be 10 to 100 times the message generation rate. Time is divided between various users to allow for multiple access, hence the term Time Division Multiple Access. For example, most people type at about 20 to 40 words per minute and the Baudot tape transmissions are sent at 60 or 100 wpm. With a packet network operating at 1200 baud approximately 6 to 10 different conversations may be on the network "simultaneously."

To reduce collisions between packets, carrier sense is implemented and to reduce repeated collisions, a random delay is allowed to pass before re-try. Thus, the Montreal Packet Net (ref.4) uses a common channel carrier sense multiple access network with 0 persistent protocol (CC CSHA 0 Persistent),

Multiple Retransmission - Multiple Routing

With a large area Packet badio Network, multiple repeater (store-and-forward) nodes extend the radio horizon and provide for multiple routing capability for netwark availability. For such a multiple repeating network, a common channel store and forward node is more appropriate than a conventinal two frequency repeater. The redundancy inherent in a large, multi-path network reduces network collapse due to a single node failure.

At some time in the future the ability to efficiently rcute packets along the best path (instead of all paths) will increase systen throughput,

Error Checking & Correction

For error reduction, packet radio net works generally use an acknowledgement (ACK) to signify successful packet reception. If an ACR is not received for a given packet, the sending node continues to retransmit and wait for an ACK for a certain number of times- The packet receiving node ascertains error-free reception by comparing the checksum with the data in the packet, This comparison may be done in software or dedicated hardware, The CRC16 (cyclic redundancy check) error checking scheme will detect all errors less than 16 bits in length, and 99% cf longer errors (ref.4).

Error correction may also be implemented using a scheme similar to error correction in dynamic RAM (Hamming codes). Additional bits are appended to each character allowing hardware reconstruction of the character without resending, if only a portion of the character was corrupted by noise.

Global Broadcasting

Global broadcasting of messages to all network users is enhanced in a packet radio network since the act of relaying is automatic, error detection and correction is inherent, and broadcast messages may be so coded in the header.

Examples of Packet Radio Networks

Currently, several groups are implementing and experimenting with packet radio networks. As outlined in ref. 1, the current major groups are:

- ARRL - American Radio Relay League, Newington, CT
- AMRAD - Amateur Radio Research & Development Corp., Washington, DC
- AMSAT - Radio Amateur Satellite Corporation, Washington, DC
- KA6M - Hank Magnuski et. al., San Francisco, CA
- VADCG - Vancouver Amateur Digital Communication Group, BC Canada
- HAPN - Hamilton and Area Packet Network, Ontario, Canada

A PACKET RADIO NETWORK FOR ILLINOIS

Current N. E. Illinois Emergency Network

For many years a Weather net" has been in operation in the greater Chicago area under the sponsorship of the N.E. Illinois Communication Association (ref. 8). The N. E. Il. Network (NEIL), WB9AGH, uses a frequency of 147.06 MHz simplex to transmit weather bulletins and ARRL bulletins at 60 wpm Baudot AFSK. Paper tape punched from the ueather wire is manually placed on a reader for transmission through a 250 watt base station with an antenna at about 80 feet. Several hundred stations continously copy the weather bulletins which during severe weather occupy almost all of the channel's throughput.

The appearance of microcomputers and the recent FCC deregulation allowing ASCII has prompted Robert Hajek, W9QBH, ARRL SEC, to consider semi-automated transmission of ASCII (in addition to Baudot).

operation Skywarn (ref. 9) currently relies on voice reports of severe weather situations to the National Weather Service. Since timly, accurate reports of severe weather, such as tornados, by on-the-scene observers, is essential in reducing damage and saving lives, an automated, error correc ting multiple access network capable of about one or two hundred mile coverage seems to be needed. What a perfect application for a packet radio network! A trained observer initiates a report automatically broadcasted to the appropriate network terminals.

Weather Telemetry

The detection, tracking, and prediction of severe weather using atmospheric radiation from severe weather (ref. 10) could be enhanced through a packet radio network (ref. 11). The needed telemetry in this system is angle, probability (ie. intensity), and time of occurrence for the sferics pulses. Using several unmanned stations scattered around the Central and Southern United States (the Rockies to the Appalachians), data could be collected, coded and sent every few minutes from the monitoring stations uia a terrestrial or satellite packet radio link. Data would be collected at a central processor to triangulate occurrences, weigh data accuracy, and infer expected severe weather trends. This information would then be made available, via the network,, to any interested party.

Other RTTY & Computer Activities in Chicago

The Chicago metropolatian area is home for many RTTY enthusiasts, some of whom use the Chicago Area RTTY Repeater System (CARRS) on 144.71 - 145.31 MHz. CARRS plans to have a mailbox system.

Many computer hobbiests who are also hams live in the Chicago area. In fact, the Computer Bulletin Board System (CBBS) was first described by two Chicagoians: Ward Christensen and Randy Suess, WB9GPM (ref. 12) .

Several other hams who have computers have commented to the authors they are individually developing a radio-accessible CBBS.

In summary, there is the need for an improved automa tic weather data network, potential amateur support exists, and the time is right for packet. radio experimentation. In fact, one logical name for this network would be CAPER - The Chicago Area Packet Emergency Radio Network since it will be a great leap (one meaning of caper) forward in message handling.

Since CAPER may connotate illicit activity, a more acceptable name could be CAPS - Chicago Area Packet radio System, Whatever the name, we look forward to the formation of a packet radio network and offer, in this paper, suggestions for such a network,

Desirable Improvements in N.E. IL Emergency Net for ASCII

As mentioned above, the NE IL Net would like to enhance the 60 wpm Baudot transmissions by offering a complimentary ASCII transmission. (It is assumed for economic reasons many stations will elect to retain their existing Baudot reception systems and will resist conversion to a packet radio network,) While surplus 110 baud Teletype model 33's are available for a modest $100 to $400 expenditure, not much is gained by transmitting at 110 baud ASCII to bardwired machines. For the present Baudot machine owners, this money would probably be better spent on a packet radio terminal node controller board. (It is a well known fact that hams are cheap, or a% least are being squeezed by inflation like everyone else-)

For the few hundred dollars an ASCII machine costs, the RTTY ham is probably better off purchasing a terminal node controller if a packet radio network is available. En fact, the VADCG style TNC boards could provide even more utility and value for the HF RTTY amateurs, A straight ASCII to Baudot and Eaudot to ASCII code and speed Converter, without packet generation, could be programmed if not already available.

Briefly, the desirable improvements proposed for the NE IL Emergency Net are:

- Two way automated communication
- Wider area coverage, especially to the West from where storms originate
- Routine ham-to-ham communication to encourage packet node construction
- Higher throughput, ie. emergency bulletins should be flashed at several times the 60 wpm speed
- Priority queueing and handling of messages
- Bulletin board service for bulleting and messages
- A regional packet radio network using protocol compatable with other area and the eventual worldwide HF and satellite AMSAT International Computer Network (AMICON)
- A protocol compatable with the VADCG TNC board for simple amateur implementation, simplified software upgrade by EPROM replacement, and protocol offloading to a dedicated processor/interface
- Simplified store-and-forward node implementation by amateurs at their home stations
- Two way link with present Baudot users (features of the packet radio network available through a standard, non-packet, Baudot AFSK VHF station)
- Operation of the statewide Common Channel Carrier Sense Multiple Access 0 Persistent network on a 2M FM simplex frequency using 1200 baud Bell 202 compatable modems, 2H FM use would be initially encouraged to ease the transition to packet, ultimately 2.20 MHz may be better due to desense, specturm crowding, etc.

Priority Queueing and Transmission

The literature on packet radio networking as applied to amateur: and HDLC VLSI chip implementation seem to ignore priority queueing and transmission, While compatability with the Intel $273, 8274, and Western Digital 1933 is commendable, we wonder if and how high priority packets should be handled? For example, a message "Tornado spotted at 3 PM in Southwest Suburb headed N.E. at 30 mph" should have priority over almost every other packet on a network and new low priority packets should not be introduced to the network until the *emergency* has passed. Possibly priority queueing is more appropriately implemented at a different protocol layer (ref. 13), ie. layer 2. Intranet, end-to-end rather than the layer 0. line control or layer 1. Intranet, node-to-node. As an example, priority queueing of certain messages is used in the SITA worldwide airline reservation network. (ref. 14).

Individual Ham Station Implementation

The above list contains advantages of packet radio implementation while at the same time illustrating the complexity of the network implementation. Fortunately, each ham does not need to be an expert computer programmer and data transmission engineer,. Just stuff the TNC printed circuit board with a pre-programmed EPROM, 8085 microprocessor, and the other support parts. Procure or build a 1200 baud modem (TU for you RTTY fans) and use almost any terminal,

Needed FCC Removal of Restrictions

It would seem to us that the FCC should be called upon to remove several restrictive regulations to promote packet radio networks.

- The requirement for a CW ID should be dropped as long as the callsign is sent in a common code eg. Baudot or ASCII
- The restriction on codes other than Baudot or ASCII should be dropped (excepting callsign) to allow synchronous data
- The requirement of a control operator present at a station engaged in digital communication should be dropped to allow unattended acknowledgements for received messages, and radio accessable Computerized Bulletin Board Systems

CONCLUSION

Packet radio networking will tremendously enhance communications in the Greater Chicago and Illinois area. *The* need for more reliable, automated emergency communication is great, We look forward to receiving comments at the Computer

Networking Ccaference and establishing diaglogue with interested amateurs.

REFERENCES

1. Borden, David W., K8MMO, and Paul L. Rinaldo, W4RI, "The Making of an Amateur Packet-Radio Network," QST, October, 1981, pg. 28-30.
2. Derr, Kurt W. "Packet Switching," Report No. EGG-2015, March 1980.
3. Magnuski, Hank, KA6M, "Packet Radio HDLC Protocol Notes," AMRAD Newsletter, September, 1981, pg. 2 and also contained in ref. 6.
4. Rouleau, Robert, VE2PY, and Ian Hodgson, VE2BEN, Packet Radio, Tab Books, Inc, 1981, catalog number 1345.
5. AMRAD Newsletter Protocol Column, May, June, July and August, 1981.
6. Letter from Rank Magnuski, KA6M, dated August 11, 1981.
7. VADCG Description in the Canadian Amateur Radio Federation Packet Radio Notes.
8. N. E. Illinois Communication Association, Inc. WB9AGH, P.O. Drawer H, Riverside, IL 60546.
9. Operation Skywarn, a voice network using repeaters on 2M and 220 MHz to relay messages to the National Weather Service of severe weather spotted by trained amateur radio operators. Since several human interveners ase used to relay a message, errors are bound to occurr in addition to time consuming delays.
10. Fergus, Richard W., W9DTW, "Real Time Analysis of Atmospheric Radiation f rom Weather Fronts," NCC '81 Personal Computing Proceedings, reprinted in September, 1198 1 AMRAD Newsletter.
11. Personal communications between Fergus & Doering, Sept. 1981.
12. Christensen, Ward, and Randy Suess, WB9GPM, "Hobbyist Computerized Bulletin Board," BYTE;, Vol. 3, No. 11, Nov. 1978, pg. 150.
13. Cerf and Kirstein "Issues in Packet Network Interconnection," Proceedings of the IEEE, Vol. 66, No. 11, Nov. 1978.
14. Schwartz, Wischa, Computer Communication Network Design and Analysis, 1977, Prentice Hall.

COMPUTER-CONTROLLED MESSAGE HANDLING

Russell D. Ward, Jr., WA4ZZU
1Ø8 Louise Avenue, #B-4
Nashville, TN 37203

Although usage of computers in Amateur Radio is slowly increasing, some study should be given to methods of promoting this utilization. This paper will show two methods of increasing computer message handling in Amateur Radio. These two methods are increased availability in the middle-term future and increased use of present capacity.

As was the case with single sideband, two things hinder widespread use of computers in ham radio: cost and complexity. Many hams are reluctant to buy equipment costing as much as an hf transceiver for which they see no immediate use and for which they have no training. Fortunately, both cost and complexity can be jointly attacked. Cost and complexity are attacked both by using equipment and skills already available and also by using dedicated microprocessor controllers. Most hams already possess the necessary input/output devices, a keyer and a TV set. These I/O devices combined with a transceiver form the basis of a simple but effective computer-based station. The missing ingredient is the computer, which can be a very simple machine. It must translate cw to ASCII, format ASCII for the TV, and properly process the incoming and outgoing messages. The most important feature of this small computer is standardized software in ROM. The operator does not initially need much computer training and can take advantage of advances in program efficiency by changing a ROM, which can be reprogrammed by clubs or the ARRL for a fee.* A very basic, low-cost, microprocessor-based message handler would be easy to construct, have an easily changed message format, and be compatible with any of the more powerful systems. Proliferation of these basic message handlers would lead to widespread use of computers in ham radio traffic work.

At the present time, there is not a lot of computer-assisted message handling. In order to steer clear of existing traffic circuits and simultaneously generate more traffic, work should be done to link several existing computer networks together. One group of hams per major city in which these existing phone-line networks are located could make their relay services available immediately. A daily relay between existing non-ham computer networks would enlarge the networks, provide message volume, and give operational experience to hams. The experience gained by years of commercial computer network operation can probably be applied to the linking of non-commercial computer networks via Amateur Radio.

*Ed. Note: The ARRL does not presently offer this service.

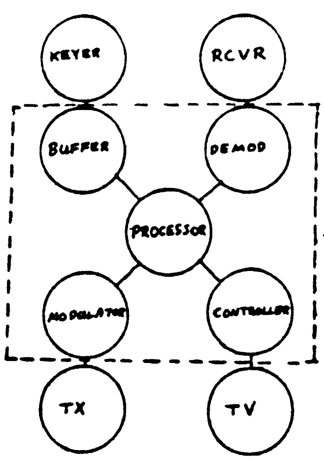


FIGURE 1.

JOE HAM NEEDS ALL INSIDE DASHED LINE.

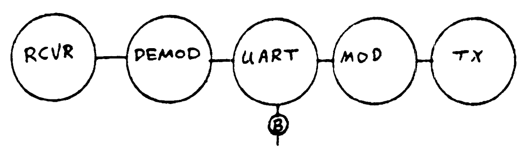


FIGURE 3.

MOD/DEMOD IS PLL AS PER QST, SEPT. 1981, P.32.

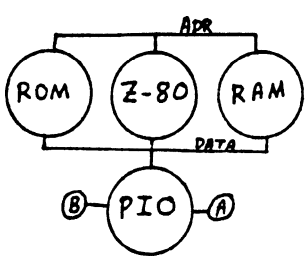


FIGURE 2.

Z80 CPU IS LOW COST, WIDELY AVAILABLE, SINGLE VOLTAGE.

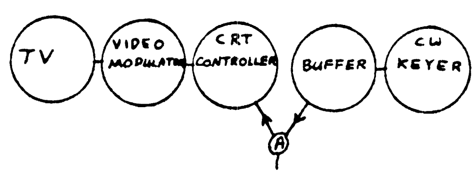


FIGURE 4.

CW TO ASCII CONVERSION VIA CPU. CONTROLLER COSTS QUITE VARIABLE.

NOTES ON CLOSING SESSION

David Engle, N6FTZ
1063 Summerwood Ct
San Jose, CA 95132

These are questions and answers at the concluding session on October 17, 1981. These questions were asked of a panel consisting of:

Dave Borden, K8MMO
Doug Lockhart, VE7APU
Hank Magnuski, KA6M
Paul Rinaldo, W4RI

Q. Concerning configuration management, how do you enforce any rule set? This question concerned the AMICON proposal for the use of the L2 channel in the Phase 3 satellites (reserved for packet communications).

A. There was no direct answer to this question, rather there was a general consensus that this could indeed be a problem. Further, this should continue to be looked at with some standards and possible enforcement mechanisms put forward for consideration.

Q. Should the use of L2 be reserved for clubs or groups of people? Should it be reserved for gateways rather than individuals?

A. There will not be any direct control of this function. It is not expected that many individuals will be willing to spend the money necessary to get into the channel. Thus, channel access will probably be run by groups. However, a caution was offered about elitism.

Q. What is the mechanism to communicate this emerging technology to amateurs? QEX? Orbit?

A. Jens Zander, SM5HEV suggested QST as the most available vehicle, especially if the European and worldwide community was to be included. Concern was expressed as to whether or not QST would handle technical matters of this depth. However, no worldwide alternatives could be suggested. It was pointed out that the IARU would need to be included in any standards setting and dissemination of those standards.

Q. The potential problem of third-party message traffic was pointed out.

A. Concern was noted, but no specific ideas were put forward.

Q. What is the 2-kHz L2 bandwidth, audio or rf?

A. The bandwidth is derived from an ssb transmitter bandwidth circa 1978-79. It is 4 kHz on center, i.e., +2 kHz.

Q. Is there any consensus on packet lengths?

A. There is no consensus at present. This varies at present with the band being used. There are many variables to deal with.

Q. Are there any plans to use spread spectrum on Phase 3?

A. No. Many problems eclipse this issue at present, e.g., legal, technical, money and other resources.

Q. Has the issue of multi-level multi-tasking software needs been addressed?

A. This issue is addressed in terrestrial systems by the Vancouver TNC board (thus removing packet scheduling from single-CPU multi-tasking.

Q. What about AMICON routing and terrestrial routing?

A. This problem has been the discussion of several informal meetings. While no solutions have been determined, the following points have been made:

a. Routing problems are still vague.
b. Call signs no longer can be used as a geographical locator.
c. There is need for 2 to 3 levels of addressing, as follows:
  1. By geographical location to Local Area Net (LAN).
  2. Within LAN, by call sign.
  3. The device at the user station.
  4. LAN addresses should be assigned centrally.
  5. The call sign table should be maintained at the LAN local level.
d. Local Area Nets should determine their own rules.
e. Entry into the backbone network should be by invitation and will utilize a rigid set of rules.
f. A one degree latitude by one degree longitude grid system could provide a basis ov a global locator. It would also fit in the internet header.
g. Another locator scheme was suggested from the floor -- the EA8EX system provides a 0.9 by 0.6 km world grid.

Q. Are priorities going to be included in any protocol standards?

A. It is an open point at this time. If it is needed it can be included.

Q. Is there any industry/ham standards going on?

A. The authors of the ham standards are looking at industry standards as they prepare the ham standards.

Q. Is a set of standards going to be put forward, and if so will any connection be allowed if it meets standards?

A. Yes. Comment on setting standards:

a. Standards are needed.
b. Standards should allow evolution.
c. Standards should start at levels of agreement wherever they may occur.

311 Stanford Avenue
Menlo Park, CA 94025
October 16th, 1981

Dear Packet Radio Enthusiast,

Thanks very much for the letter of inquiry which you sent me. The response I've received to the initial publicity about the packet repeater has been very enthusiastic, and I have been deluged with requests from hams, both locally and from various points around the country, for more information about the repeater, for schematics, for listings, specifications, modems, proms, HDLC chips, Vancouver boards, and for talks at clubs. Needless to say, all this activity, plus continuing development on the packet hardware and software, has kept me very busy, and I apologize for any delay in responding to your letter. Let me bring you up to date on what has happened, or is happening, since the initial announcement of the repeater, which went on the air in December of 1980.

In the early months of this year, the packet repeater was operating out of my residence, and was still an experimental machine, Since then, we have installed a couple of upgrades to the control software, we have used a better CPU card, increased the power level, moved the repeater to 700 feet elevation, and integrated its operation to be 100% compatible with the protocol used by the Vancouver Digital Communications Group (VADCG). The repeater has changed from being a laboratory curiosity to a major Bay Area repeater heard from Berkeley to south San Jose, and the user community has grown from a couple of stations to a network of some 30 users, The packet system here now has a mailbox on-line 24 hours a day, several on-line personal computers, and network links (courtesy of a commercial packet network) to the other active packet radio centers in Washington, Vancouver and Ottawa. We have also just installed an HF port on 20 Meters, and are beginning some experiments aimed at establishing a connection with AMRAD in Washington and with equipment located at W1AW.

Most of the original packet radio experiments were done in Canada (in part due to the Canadians" pioneering communications spirit, and in part due to less restrictive regulations up there), and three main centers were at work: Montreal, Ottawa and Vancouver. The technology employed by each of these groups differed, and each approach has its own merits. My thinking and ideas very closely paralleled the work started by Doug Lockhart, VE7APU, and so I can best report on what is happening with groups which have adopted HDLC (High-level Data Link Control) framing as the basis of their protocol. The HDLC/SDLC frame is a new, universally accepted standard in the data communications industry, and Doug and I feel it offers a good starting point on which to build a packet-radio network. As it turns out, groups in Washington D.C., Los Angeles, El Paso, Denver, Sacramento, Tuscon, and Hamilton have also taken up this technology, and it is likely that we already have a sufficient number of people using this technique that it will become the defacto standard in the amateur radio community.

It would be impossible for me to completely describe the protocol and equipment being used in this letter, so I will briefly cover some of the topics and give you some pointers on where to find additional information. As you might guess, this is a new area for amateur radio, and tutorial material and handbooks are not easily found. Many issues and problems remain to be discussed, and there is opportunity to make substantial contributions to the state of the art.

The Protocol - The basis for our technology is the HDLC frame, which is simply a way of encapsulating a series of bits into a message block. This type of message framing offers a high degree of error detection, data transparency, NRZI encoding, and a comprehensive set of standards for its use in a point-to-point protocol. Source documents for HDLC/SDLC and protocol are the following:

IBM Synchronous Data Link Control, General Information, GA27-3093 (This manual is available through your company's IBM comp center or over-the-counter at most major IBM offices.)

Advanced Data Communications Control Procedures, ANSI X3.66-1979 (This is the American standard or HDLC. Write ANSI, 1430 Broadway New York, New York, 10018 for current price information.)

For easier to get references, most recent books on data communications have chapters on "bit-oriented-protocols". These are some sources available:

IEEE Transactions on Communications, COM-28 No. 4, April 1980 (Special issue on Computer Network Architectures and Protocols)

IEEE Proceedings,, Vol. 66, No. 11, pp 1301-1588, November 1978 (Special issue on Packet Communications Networks)

Technical Aspects of Data Communications by John E. McNamara, Digital Press (See Chapter 19 on Bit Oriented Protocols)

Communications Architecture for Distributed Systems by R.J. Cypser, Addison Wesley (See Chapter 11 on Data Link Control)

NCC (National Computer Conference) Proceedings, 1975, AFIPS Press (Many excellent articles on packet radio concepts)

Doug Lockhart's initial implementation of the protocol is a subset of the IBM SDLC standard. The enclosed notes document the type of frames used.

HDLC Chips - Quite a few manufacturers are now making HDLC/SDLC VLSI chips. Of the half-dozen or so available, only two have all the hardware required for easy use in an amateur radio environment. The special features required are NRZI encoding and an on-board digital phase-locked loop for timing recovery. The two chips which have been used by amateurs are the Intel 8273 and the Western Digital 1933. The Intel component data sheet has a brief summary of HDLC framing. In quantity purchase the Intel part is about $35, and the Western Digital part is $30.

Repeater Hardware - The repeater hardware is based on STD bus cards. The STD bus uses 56-pin 4 1/2" x 6 1/2" cards and is very popular for industrial process control applications. There are now many manufacturers supplying cards for this bus, and the repeater uses the Z-80/CPU-2 card from Mostek, which costs about $195. There is no reason why S-100 Z80 cards could not be used for a repeater. The STD card is very compact, and does not have unneeded extra circuitry which is typically found on more versatile personal system CPU cards, A WD1933 chip was breadboarded onto a Vector STD board, and one additional card to control the transmitter is all that was required. The software, written in PASCAL/Z and assembler fits into two 2716 EPROMS, and 2K bytes of RAM memory is required. The repeater can store up to 8 128-byte packets. The software and schematics will be available from me for a fee. Write or call if you are seriously interested in duplicating this equipment. The radio is an FT202 handheld transceiver with a 20 Watt power amplifier. Since this repeater runs simplex, problems associated with full-duplex repeaters (desense, etc.) are not encountered, and exotic remedies, such as duplexers, dual antennas, and so forth, are not required.

The repeater serves to increase geographic range due to its advantageous location, it digitally regenerates the packet, providing all stations with a uniform signal, it selectively repeats only those packets addressed to it, allowing the possibility of multiple repeaters on the same frequency (an advantage instead of a curse!), and its beacon and packet-repeat facilities allow stations to do full-loopback testing, an invaluable resource in bringing up new equipment and checking out hardware/software modifications.

The VADCG Terminal Node Controller - Doug's group designed a low cost, 8085 based circuit board which uses the Intel 8273, and which has proven to be a very easy way to get started in packet radio. The hardware/software does an excellent job and many are in use here in SF. Most people ask "Why can't I put this software onto my personal computer?" First, HDLC T/O interfaces are not commonly found on personal computers. Next, the VADCG TNC acts as a dedicated peripheral controller' handling the radio link protocol, and thus off-loads this responsibility from the home system. The personal computer can go off reading/writing its disks while packets are coming in over the air, Handling two full-duplex I/O paths at high speeds is a non-trivial programming job and is above the skill level of many amateurs. The current program is about 30 pages of assembly language, and is not particularly easy to transport to other 8080/Z80 environments because of the interrupt handling required. We have managed to support a wide variety of different home systems with one software package, and an upgrade to the program requires replacing some EPROMs, not rewriting the software for 30 different DOS's. Thus, for newcomers I would highly recommend buying a pair of TNC's, as the cost of the bare board is only about $35. Total cost fully loaded is around $200. The software is available from several sources around the country. The TNC could also be programmed to be a packet repeater, with relatively simple modifications to the existing code.

Modems - The Bell 202 modem is a voice frequency 1200 bits-per-second modem which is ideal for radio work at 2 meters and which can now be found at reasonable prices through various surplus outlets, or which can be fabricated using EXAR chips or op-amps. This modem has become standard for amateur radio use at this Baud rate. At one time I had a good supply of these modems for $75-$90, but I'm totally sold out at present. There are as yet no standards for higher speeds or direct digital modulation of an RF carrier. This is an area which needs more experimentation.

Newsletters - Our group does not have a newsletter, However, the following organizations periodically publish very fine newsletters which deal with packet radio:

AMRAD, 1524 Springvale Avenue, McLean VA 22101

VADCG, 818 Rondeau Street, Coquitlam, B.C., Canada, V3J 5Z3

Hamilton Area Packet Net, Stu Beal, Editor, 2391 Arnold Crescent, Burlington, Ontario, Canada, L7P 4J2

Reference Material - Some recent articles and publications have addressed themselves specifically to packet radio in an amateur radio environment:

Borden and Rinaldo, "The Making of an Amateur Packet-Radio Network.", QST, Volume LXV No. 10, October, 1981, pp. 28-30.

Rouleau and Hodgson, "Packet Radio", TAB Books, Blue Ridge Summit, PA, ISBN o-8306-1345-5, June 1981

Proceedings of the ARRL Amateur Radio Computer Networking Conference, Volumes I & II, October 16-17, 1981. Write AMRAD or ARRL.

Networking - Most active packet users agree that the real driving force behind packet is the ability to interconnect geographical areas through networks. Electronic mail, digitized imagery, networking games, computer conferences and other unthought of applications remain to be explored. The next year should bring some heavy-duty activity in the areas of defining network protocols and standards, and maybe even some interconnection between a few of the centers of packet radio activity. A conference to discuss these issues was recently held in Gaithersburg, Md., and from the number of people in attendance, and from the variety and depth of papers presented, it seems to me that a major new area in amateur radio has been opened.

Thanks again for your interest. See you on the net.

Best regards,

Hank

Hank Magnuski, KA6M

STEPHEN E.	BACH	AA4B	RT 2 BOX 89	SCOTTSVILLE VA	24590-9512	804-286-3466
RICHARD	BARTH	W3HWN	11523 CHARLTON DR	SILVER SPRING MD	20902	301-681-7372
KEN	BESS	K4EVH	11160 SAFFOLD WAY	RESTON VA	22090	703-437-4360
HARRY G.	BLUESTEIN	N6TE	5533 MOONLIGHT LANE	La Jolla CA	92037	714-454-1098
DAVID	BORDEN	K8MMO	RT 2 BOX 233B	STERLING VA	22170	703-450-5284
STEPHEN A.	BRENNER		8400 JEB STUART ROAD	ROCKVILLE MD	20854	301-762-4845 TTY
JIM	CARD	K3VJH	2219 ROSS COURT	SILVER SPRING MD	20910	301-565-2219
ROBERT J.	CARPENTER	W3OTC	12708 CIRCLE DRIVE	ROCKVILLE MD	20850	301-762-5838
GARY C.	CHATTERS	WA9ZZZ	9110 8TH STREET	SEABROOK MD	20706	301-459-8143
DR. THOMAS A.	CLARK	W3IWI	6388 GUILFORD ROAD	CLARKSVILLE MD	21029	301-286-3113
BOB	CLEMENTS	K1BC	4 PAUL REVERE ROAD	LEXINGTON MA	02173	617-861-1071
DEN	CONNORS	KD2S	4708 W. WICO HORSE DR.	TUCSON AZ	85741	602-744-9306
MARTIN	DAVIDOFF	K2UBC	13803 MANOR GLEN ROAD	BALDWIN MD	21013	
THOMAS L.	DAVIS	WB6PYY	2480 DEXTER #5	SEATTLE WA	98109	206-285-3388
JOHN S.	DEGOOD	WB3GNB	P.O.BOX 294	YORKLYN DE	19736	302-239-6034
ROBERT J.	DIERSING	N5AHD	4129 MONTEGO	CORPUS CRISTI TX	78411	512-852-3196
RICHARD W.	DOERING	WA6CFM	1037 CORNISH DR.	SAN DIEGO CA	92107	714-222-7267
EDWARD A.	DUNLOP		P.O.BOX 4568	NEWARK DE	19711	302 453-8175
JOHN	DuBois	W1HDX	873 HILL ROAD	BOXBOROUGH MA	01719	617 263-7004
DAVID	ENGLE		1053 SUMMERWOOD CT.	SAN JOSE CA	95132	408-251-2910
JIM	FENTON	K1OH	655 S. FAIR OAKS AVE #H-115	SUNNYVALE CA	94086	408-737-1416
GEORGE S.	GADBOIS	W3FEY	141 MAPLE LANE	LANCASTER PA	17601	717-397-0158
PAUL	GALLI	SM5DYW	VAESTANAAG 20	LINKOPING SWEDEN	5-58235	013-147-7839
GARY	GARRIOTT	WA9FMQ	6913 COLUMBIA DR.	ALEXANDRIA VA	22307	703-768-2606
FORREST R.	GEORGE	W9SKD	16219 S. GEORGE CT.	PLAINFIELD IL	60544	
GORDON	HARDMAN	KE3D	9314 CHERRY HILL RD. #923	COLLEGE PARK MD	20740	301-441-2946
NORM	HECK	W3NN	9309 WISCONSIN AVE	BETHESDA MD	20814	301-530-6677
BILL	HENRY	K9GWT	616 W. CHURCH	CHAMPAIGN IL	61820	217-359-8342
WILLIAM T.	HODSON	W0HNF	15 CRESCENT LANE	COLO SPRINGS CO	80904	303-632-3988
J.	HOLIBAUGH	K3WPG	4715 FALCON ST.	ROCKVILLE MD	20853	301-949-7861
SAM	HUNT		4711 OLLEY LANE	FAIRFAX VA	22032	703-323-7627
STEVEN C.	JOHNSON	WB3IRU	13826 DOWLAIS DR.	ROCKVILLE MD	20853	301-460-4766
ED	KALIN	K1RT	83 KENMORE ROAD	BLOOMFIELD CT	06002	
PHIL	KARN	KA9Q	1740 PADDINGTON	NADERVILLE IL	60540	312-961-9054
GEORGE	KINAL	W3HPF	636 S.C. AVE	WASHINGTON DC	20003	202-546-7270
JAN A.	KING	W3GEV	4020 CHANEY COVE CT.	DUNKIRK MD	20754	301-855-7908
R.	KOPLOW		BOX 368	ROCKVILLE MD	20850	301-340-0779
LAWRENCE	KOZIEL	K8MU	42509 PARKHURST	PLYMONTH MI	48170	313-420-0786
GEORGE	LEMASTER	WB5OYP	7809 BEARD COURT	FALLS CHURCH VA	22043	703-573-1226
BEACHAM	LEONARD	W4RJ6		WINSTON-SALEM NC	27103	919-724-7350
DOUGLAS	LOCKHART	VE7APU	29 SHAMOKIN DR	TORONTO ON CANADA	M3A 3H7	416-441-2417
SCOTT	LOFTESNESS	W3VS	20324 HIGHLAND HALL DRIVE	GAITHERSBURG MD	20879	301-963-3473
HANK	MAGNUSKI	KA6M	311 STANFORD AVE	MENLO PARK CA	94025	415 854-1927
LOUIS A.	MAMAKOS	WA3YMH	349 MARGANZA S.	LAUREL MD	20707	301-776-6517
T.P.	MATHEWSON	W4FJ	1525 SUNSET LANE	RICHMOND VA	23221	
JOHN WALL	MILLER	W4FRW	6921 PACIFIC LANE	ANNANDALE VA	22003	703-256-5702
RONALD G.	MINNICH		315 MERCER MILL ROAD	LANDENBERG PA	19350	215-274-8678
JOHN	MONTAGUE	W0RUE	939 ARBOR	MAHTOMEDI MN	55115	612-623-9976
KEN	NICHOLS		213 DAYTON ROAD	GAITHERSBURG MD	20877	301-977-5405
DR.W.P.	PALA	WB4NFB	5829 PARAKEET DRIVE	BURKE VA	22015	703-323-8245
GEORGE B.	PEARSE	KA1DF	1370 MAIN STREET	MILLIS MA	02054	617-366 7-44
ROLAND S.	PHELPS	WA3UZJ	7607 BLACK ROAD	THURMONT MD	21788	301-271 443:
JOHN G.	PRONKO	W6XN	230 HAWTHORNE AVE	LOS ALTOS CA	94022	415-941-6988

TED	RABENKO		4405 MIDSTONE LN FAIRFAX VA	22033	703-378-4358
MILTON GWYN	REEDY	W1BEL	3 BAYAU TRAIL MEDFORD NJ	08055	609-654-1184
TEDD	RIGGS	KC4YN	8402 BEREA COURT VIENNA VA	22180	703-573-5067
PAUL L.	RINALDO	W4RI	1524 SPRINGVALE AVE MCLEAN VA	22101	703-356-8918
STEPHEN E.	ROBINSON	W2FPY	47 SERPENTINE ROAD RINGWOOD NJ	07456	201-835-1152
NAN	SANDERS		404 PARK STREET S.E. VIENNA VA	22180	703-281-9390
ELTON	SANDERS JR.	WB5MMB	404 PARK STREET S.E. VIENNA VA	22180	703-281-1298
ERIC L.	SCACE	K3NA	10701 FIVE FORKS ROAD FREDERICK MD	21701	301-293-2448
CHARLES	SCHENCK	W1EH	83 MAIN ST. #70 NEWINGTON CT	05111	203-666-1103
WILLIAM	SCHOLTZ	W3HXF	9407 GOOD LUCK ROAD SEABROOK MD	20706	301-577-7353
KEN	SHEERS	WA3VIW	5425 ENDICOTT LANE COLUMBIA MD	21044	301-730-0720
THOMAS J.	SHINAL	W4HCP	P.O.BOX 166 WATERFORD VA	22190	703-882-3588
JIM	SKOOG	WD0EEL	1265 N.W. 7TH ST. BOCA RATON FL	33432	305-395-4575
GEOFF	SMITH	WB3ANZ	4101 FAIRFAX AVE LANDOVER HILLS MD	20784	301-341-5448
ROBERT H.	SMITH	KA1VF	1 WATER ST. APT 514 HAVERHILL MA	01830	617-374-8371
PETER A.	STARK	K2OAW	P.O.BOX 209 MT.KISCO NY	10549	914-241-0287
DAVID B.	STEVENS		869 N. COLLEGE PARK ST. COLLEGE PARK MD	20740	301-577-1300
RALPH	STIRLING	KC3F	6716 COCKERILLE AVE TAKOMA PARK MD	20912	301-270-4854
RICHARD	SUDKAMP	WA3URU	9812 GENEVA LANE FREDERICK MD	21701	301-662-4898
ARTHUR	TELLER	KB4ZJ	6511 TUCKER AVE MCLEAN VA	22101	703-827-0270
HUGH	TURNBULL	W3ABC	6903 RHODE ISLAND AVE COLLEGE PARK MD	20740	301-927-1797
CHRIS	WACHS	WA2KDL	72 TIERNAN ST. ROCHESTER NY	14612	317-495-1311
ROBERT	WATSON		121-D WATKINS MILL ROAD GAITHERSBURG MD	20760	301-963-8360
KEAN	WERNER	WA3THU	13620 ELLENDALE DR CHANTILLY VA	22021	703-378-4110
DANA E.	WHITLOW	K8YUM	1680 S. STATE ANN ARBOR MI	48104	313-994-0159
PERRY	WILLIAMS	W1UED	12 WEST DISTRICT ROAD UNIONVILLE CT	06085	203-673-2810
CHAS P.	WILSON JR.	K4CAV	4101 SUMMERGLEN CT. GREENSBORO NC	27406	919-294-3480
SCOTT W.	WOOD	KA3AJL	19106 N. KINDLY CT. GAITHERSBURG MD	20879	301-869-0622
DAVID	WOODALL		12500 CHRONICAL DRIVE FAIRFAX VA	22030	703-830-2261
DAVID R.L.	WORTHINGTON	WA6WLE	P.O.BOX 274 LA HONDA CA	94020	415-747-0021
JENS	ZANDER	SM5HEV	DEPT. OF EE LINKOPING UNIVERSITY LINKOPING SWEDEN	S-58183	013-176-6700
RICHARD	ZWIRKO	K1HTV	12509 RANSOM DR GLENN DALE MD	20769	301-464-2133
PETER	de BRUYN	W3EXP	12907 CROOKSTON LANE A4 ROCKVILLE MD	20851	301-881-8330

Second ARRL Amateur Radio Computer Networking Conference

March 19, 1983
San Francisco, California

Coordinators:
Hank S. Magnuski, KA6M
Paul L. Rinaldo, W4RI

Committee:
Stu Neblett, K6VCO
Bob Reiling, W6JHJ
Curtis Spangler, N6ECT
San Francisco Radio Club

Hosted by:

Amateur Radio Research & Pacific Packet Radio Society
and Developement Corp.

Den Connors, KD2S
PACSAT Project Manager
Radio Amateur Satellite Corporation - AMSAT

AMSAT has begun the design and development of a new form of Amateur Satellite. The PACSAT series of satellite systems has as a design goal total global access by all hams to a store-and-forward packet radio message handler.

Introduction

AMSAT is proposing the design and prototyping of a satellite-based experiment for advanced digital packet satellite communications experiments. This system, called PACSAT, will use internationally-allocated Amateur Radio Service frequencies. The PACSAT system will connect a grid of ground-based amateur radio local area networks in the United States and many other countries via a common store-and-forward packet repeater operating in the Amateur Satellite Service.

This paper details the reasons behind such a satellite. Following the design concepts, a description of the entire system is given, and a list of technical parameters for each of the defined subsystems is shown. The current outline of tasks and scheduling follows, with a description of the efforts of groups already engaged in the initial design effort.

The PACSAT Concept

The Amateur Radio communities in the United States and other countries are currently experimenting with digital networks on radio channels. These networks are using techniques already in place on the national telecommunication networks known collectively as packet networking.

Packet radio systems have a set of benefits unusual in present amateur radio systems. Large numbers of stations may share a common frequency, and use multiple access packet techniques to multiplex several sets of users in the time domain; very high spectrum utilization is accomplished by keeping all of these users on the same channel. A second benefit of the single shared channel is the ability to find all other users of the packet radio system. No searching of a wide band of frequencies is required; connectivity is maximized. The need of multiple access techniques to detect successful transmissions yields a third benefit, that of reliable transmissions. Any message that arrives at destination has had its data integrity checked. This inherent reliability may well open a series of possibilities for improving emergency traffic handling, one of amateur radio s most important aspects.

As experiments continue on ground-::;;," packet radio local area networks, a new of satellite is being considered to handle linking of both individual ground stations and local area networks. The PACket radio SATellite (PACSAT) system is designed to provide a store-and-forward digital repeater which is available to all groups around the world for fully global network coverage. The satellite provides this coverage by occupying a low-earth orbit (LEO), which has several benefits. The close proximity of passage, relative to geo-synchronous satellite, allows easy access, with good link margins ". There are thousands of amateur radio earth stations that are already configured to operate on this class of satellite. Additionally, proper choice of orbital parameters allows a sun-synchronous orbit, where passage of the satellite occurs at the same times each day, providing an easy means of scheduling transmissions. This orbit then provides both 100% global coverage and very fair access, and creates a powerful new use of a well-known class of amateur satellite.

There are several purposes for providing such a system in the Amateur Satellite Service. PACSAT will provide a wide-availability vehicle for advanced experimentation, and a prototype system for a new class of satellite service involving reliable transmission of data to remote sites and isolated users regardless of location. Several internationally-based organizations have expressed interest in just such capabilities, and this gives AMSAT the oppurtunity of spear-heading a potentially major new push in low-cost satellite systems, much in consonance with the F.C.C. charter for the Amateur Radio Service as a "proving-grounds" for new technology. The Volunteers in Technical Assistance (VITA), a non-profit firm dedicated to advancing the level of technilogical expertise of less-developed countries, is actively pursuing the coordination of such vanguard activities, and is working directly with AMSAT on the PACSAT development.

Other benefits result from the use of digital techniques. Considerable improvements may be made to emergency communications, as reliable, high availability links compatible with global mobile and portable radio service requirements are provided. Additionally, a spin-off benefit for AMSAT itself is the attracting of the new, computer-aware members of the Amateur community into the Amateur Satellite Service.

A possible third set of benefits may be spun off the PACSAT system indirectly; the opportunity exists for designing new types of low-cost satellite launching and propulsion systems as a part of this next generation of Amateur spacecraft.

Such a system would provide a number of functions. In addition to the primary use as a world-wide store-and-forward link, or "flying mailbox", the PACSAT experiment could provide real-time regional linking (standard LEO amateur mode). As mentioned, both local network concentrators (gateways) and individual users could access the satellite. Finally, the system would provide the mechanism for advanced testing of network systems concepts, hardware, software and protocols to be used by packet radio networks in the future.

Other benefits result from the use of digital techniques. Considerable improvements may be made to emergency communications, as reliable, high availability links compatible with global mobile and portable radio service requirements are provided. Additionally, a spin-off benefit for AMSAT itself is the attracting of the new, computer-aware members of the Amateur community into the Amateur Satellite Service.

A possible third set of benefits may be spun off the PACSAT system indirectly; the opportunity exists for designing new types of low-cost satellite launching and propulsion systems as a part of this next generation of Amateur spacecraft.

Such a system would provide a number of functions. In addition to the primary use as a world-wide store-and-forward link, or "flying mailbox", the PACSAT experiment could provide real-time regional linking (standard LEO amateur mode). As mentioned, both local network concentrators (gateways) and individual users could access the satellite. Finally, the system would provide the mechanism for advanced testing of network systems concepts, hardware, software and protocols to be used by packet radio networks in the future.

PACSAT System Description

PACSAT is an extraordinarily complicated system, rather similar in complexity to the Phase III spacecrafts. In addition to a.11 of the

required satellite support subsystems on board the spacecraft, there are two experimental packages, each consisting of multiple uplink channels, common downlink channels, and modems, coder/decoder link-access devices and control microprocessors with interfaces to a common satellite message processing unit (system control unit, or SCU). As if that isn't bad enough, the rigid packet environment demands structured ground stations with all of the familiar hardware (less the directional antennas, as we shall see), and perhaps several micropressors for handling both the data stream and the automatic station control functions. The age of the microprocessor is upon Amateur Radio.

To ease the burden of trying to understand the whole system, PACSAT can be broken up into subsections, each with well-defined interfaces to other sections. A description of each section or interface follows. Please note that, although the conceptual design has been finished, many design groups are hard at work coming up with the specifications for their parts of the overall system, so that nothing below can be consrued to be the "final word".

Spacecraft

As mentioned, the orbit of PACSAT would be sun-synchronous, that is, appearing at the same time each day. UoSAT/OSCAR 9 has this type of orbit, and displays this property. Additionally, such an orbit guarantees at least two passes per day will be seen by ALL corners of the Earth.

The PACSAT satellite system may be broken into the spacecraft itself, and the experimental packages. The interfaces are defined to be spacecraft/ experiment and spacecraft/ground station.

Two options are available for placing PACSAT experiments into space. The possibility of riding the packages inside of a spacecraft built primarilly for other purposes exists, and allows the PACSAT design team to avoid the additional complexity of designing and building all of the required subsystems. AMSAT has looked in particular at tne future launch opportunities available on the Conestoga-series of launch vehicles to be provided as a commercial venture by Space Systems Of America, Incorporated. SSI will be launching payloads directly into low-earth orbit, providing a mechanism for direct injection of PACSAT into its final orbit without requiring on-board propulsion systems in the satellite.

A second opportunity is more in line with AMSAT's traditional method of designing the satellite "from the ground up", and will likely provide many more opportunities for future launches. The Space Transportation System (Space Shuttle) has the option of carrying into space sets of three "Get-Away Special" canisters, or GAS cans. Although these cans have traditionally been reserved for inexpensive access for experimenters who did not require throwing their experiments into space, recent discussions with NASA have shown promise for using such a can as a launch opportunity. A satellite would be placed inside the can, with a mechanism in place to allow the Shuttle crew to remotely open the lid and push tne spacecraft into the void (hopefully after opening the Shuttle bay doors).

This new opportunity has two tremendously useful aspects: GAS can opportunities are CHEAP ($10 000) and potentially plentiful. The drawbacks are the requirements of building such spacecraft as would be required to fit into the can, and providing a propulsion mechanism for altering the very low orbit into which the Shuttle would place the unit, so that a final, more stable orbit would be available. As it happens, there are active international AMSAT groups that are very excited with the possibility of providing both spacecraft and propulsion.

The University of Surrey spacecraft design team (UoSAT) has expressed an interest in continuing their advanced low-cost spacecraft design and construction projects, and view PACSAT as an excellent opportunity for using their integration expertise, and for providing a vehicle to carry other experiments of interest to their group.

The AMSAT/DL team at the University at Marburg, West Germany, has been discussing the possibility of providing an innovative spacecraft engine which would be ideal for such a craft as PACSAT - a steam engine, not unlike those first designed by Hero in ancient Greece. The mechanism for generating steam in space is not difficult, and impinging sunlight on external water tanks could provide a large part of the energy required to heat the water. Heating coils electrically powered in the area of the super-heated steam nozzles would finish tne heating job. Although this concept seems a little far-fetched at first, calculations prove the amount of water required to alter the orbit of PACSAT is quite modest. Further, the ever-present problem of safety to the Shuttle crew is very much reduced by having spacecraft with extremely non-volatile fuels such as water! Gradual pushes from the steam nozzles at opposites sides of the orbit will nudge PACSAT into its final orbit, and residual water could be used to furtner occasionally alter the orbit to keep it in a sun-synchronous plane.

PACSAT Communications Experiment Package

Each of two packages will contain a set of uplink and downlink channels with associated analog and digital hardware. Current designs are targetted for typically four uplink channels, each dynamically configurable with respect to data rate. One high-speed downlink channel will be used to support the uplinks, and to provide control over the smart ground stations. For an excellent review of the design effort for tne modulation techniques and access modes of these channels, see the paper "Modulation and Access Techniques for PACSAT" by Phil Karn, which is included in these proceedings.

Supporting these communications channels will be a series of filters, oscillators and amplifiers, along with microprocssors and buffer memory for channel control and support of link access protocols. These processors, with perhaps one or two channels per processor, will allow the demodulators chosen to be both adaptive in data rate and frequency agile.

The set of packages will have a common system controller and main memory unit (RAMUNIT). The software to support the higher-level protocols and application programs to be resident in the SCU will be loadable from the ground, a technique now common in tne Amateur Satellite Service. A memory package in the megabyte range is being investigated.

Spacecraft/Experiment Interface

The spacecraft will provide the envvironment for PACSAT, including power, antennas and shielding from the extremes of space. A separate pocessor will handle the spacecraft's housekeeping functions, and separate communications channels will be available for satellite command. Standard interfaces will define nd stations will be fairly complicated, requiring smart controllers to handle the re uirements of frequency agility in the transmitters, and of linking, networking and presentation control.

To allow users to ease into packet radio satellites, a gradual upgrade pa%h is to be provided for PACSAT use. A required piece of equipment will be the modem, which will include a modulator, demodulator and pass-through path for transmitter push-to-talk and frequency control. This modem will be capable of operating as a stand-alone modem attached to one of the current types of packet radio terminal node controllers (TNC).

Operation of the TNC and modem pair with a standard set of 440-MHz transmitter and L-meter receiver will allow operation at 1200 bauds. Higher speed operation will require a separate rf deck, witn direct access to IF strips. Speeds of up to 9600 bauds are planned.

A final touch would be a custom TN, specifically designed for this system, and allowing direct interface to otner TNCs for ground-based internetwork linking.

It should also oe noted that conservative link margin calculations have shown that, with modest transmitter power on board the spacecraft and standard powere available to ground stations (around 25 watts,), the requirement for having directional antennas is not necessary. Simple gain verticals like 5/8th whips on 440 and 2 will probably be quite adaquate, especially at lower baud rates like 1200 bauds.

Spacecraft/Ground Station Interface

The PACSAT Project intends to use omnidirectional antennas on two of the most popular vhf/uhf bands, in a mode which will be familiar to Phase III users. Uplinks will be available at around 435 MHz, and choice of the proper channel will be made by the ground station controller, following the command requests of tne satellite. The common downlink will appear at the edge of the Amateur Satellite allocations, probably around 145.806 MHz for one package and 145.994 MHz for the other.

The modulation technique, synchronization requirements, encoding mode and related parameters are to be determined, based on experiments to be performed by two different design teams this spring. It is assumed that either differentially-encoded phase-shift keying or minimum shift keying at rates in the 1200 to 9600 baud range, perhaps adaptively availble, are the most likely candidates.

The link-level access protocols, that is, the addressing and error detection schemes are planned to be corpatible with the AX.25 Amateur packet radio protocol standard. This protocol has already been implemented by several groups, and is a de facto AMSAT standard for all currently-planned packet satellite efforts.

The network protocol will probably support AX.25 network-level protocol, and perhaps also less corplicated (and less reliable) "datagram-type" protocols as well.

The memory interface between ground stations and the on board RAMUNIT will be little more than a virtual disk drive. with a very noisy connecting link. On top of tnis protocol will lie an applications program which will provide a number of message and file services. Experimenters will be provided with lower-level accesses to the system where such access does not significantly disrupt normal use of tne system.

PACSAT Project Status

The final conceptual review meeting was held in February 1983, and several of tne design groups attended, including representatives from both VITA and the University of Surrey. Many of tne more sophisticated concepts were thrown away to provide an easier target for scheduling. There will be a set of subsystem design meetings at this conference, and further meetings to be held later this spring. Negotiations are currently underway with the candidate launch agencies and design support groups.

System design is likely to be completed early this summer, with deliverable items to be integrated and tested this fall. Following critical design review meetings, spacecraft-ready subsystems will be prepared and shipped to the integrating agancy by next spring. Such a schedule would allow AMSAT to take advantage of possible launch opportunities as early as late 1984. Slippages will be inevitable however, and more'realistic times will in general coincide with the more likely target launch dates, early 1985 to 1986.

The project now has the support of twelve differennt design groups from four different countries, but is still in need of qualified hardware and software designers to help review all aspects of tne current design, and provide needed manpower with several of tne more important subsystems. PACSAT is an all-volunteer effort, and will require careful evaluation by the general user community during its initial phases to confirm design parameters and provide guidance in the utility of tne various modes. It is hoped that this "system will not only provide many services which are forecast for the digital fututre of ham radio, but also create a whole new set of users and uses vet to be imagined.

Terry Fox, WB4JFI
Vice-President, AMRAD
1819 Anderson Road
Falls Church, VA 22043

Abstract

This paper contains the latest draft of the AX.25 protocol Specification. This is the first public release of this draft. Earlier drafts have been given to specific inviduals for comment and as a reference for software development. Changes should be expected. Please check the AMRAD Newsletter for announcements of later versions.

History

Over the years there have been several protocols suggested for use at layer 2 of the ISO Open System Interface Reference Model (OSI-RM) over Amateur Radio. The one system that has been in use is based on the IBM SDLC protocol and it has been working as far as it went. One of the immediate problems that came up with SDLC was that its address field of SDLC is very limited (being one byte long) causing problems if there are many amateurs on at a time.

Trying to come up with a protocol that everyone would agree to seemed like an almost impossible task a year ago. What we at AMRAD decided to do was to go over the various protocols in use or available to the amateur, figure out the best and worst parts of each protocol and see if the protocol could be "enhanced" to work properly over the amateur radio enviroment. After reviewing the various protocols around and talking with people in the computer networking industry, we decided to push the X.25 standard modified to allow a larger address field. At about this time, a group of amateurs in New Jersey were coming to the same conclusion, so about mid-June of 1982 the two groups got together and after two weekends came to an "understanding" on a level 2 protocol. The most delicate part of the negotiations between the two groups concerned the name to be given this protocol. In order to not step on anyones toes, it was decided to call the protocol AX.25, which stands for Amateur X.25.

The next step in the evolvement of AX.25 was that in October of 1982, AMSAT hosted a gathering of some of the leaders in amateur packet radio. AMRAD was at the meeting, along with representatives from TAPR, SDAPR, AMSAT, and PPRS. Three days of intense discussion followed, and an agreement was finally reached on a nation-wide compatible protocol. AX.25 was then modified to be compatible with this new protocol (basically the only major changes were an additional extension of the address field, and the addition of a Protocol Identifier, or PID field).

The rest of this paper will describe the basics of the AX.23 level 2 protocol.

AX.25 Layer 2 Protocol Specfcation

This protocol conforms with the ISO Recommendations 3309, 4335 (including DAD 1&2) and 6256 high-level data link control (HDLC) and uses some terminology found in these documents.

This protocol also conforms with ANSI X3.66, describing ADCCP, balanced mode.

This protocol is written to work equally well in either half- or full-duplex amateur radio environments.

This protocol has been written to work equally well for either Point-to-point connections, or connections made thru a larger device, such as a metropolitan network controller (MNC).

This protocol does allow the establishment of more than one layer 2 (link layer) connection per device, if the device is so capable.

This protocol also follows in principle the CCITT X.25 recommendation, with the exception of an extended address field and the addition of the Unnumbered Information (UI) frame.

Most layer 2 protocols assume that one large device (generally called a DCE, or data circuit-terminating equipment) is connected to several smaller devices (usually called a DTE, or data terminating equipment). AX.25 assumes that both ends of the link are balanced, thereby eliminating the two different classes of device.

Frame Structure

Level 2 packet-radio transmissions are sent in small blocks, called frames. These frames are made up of smaller parts, called fields. Fig. 1 shows how the three types of frames are made up. Fig. 1 shows the frames in the same bit order that most packet articles show them. Unfortunately, this method has led to some confusion, since the least-significant bit (LSB) is to the left rather than to the right, as most people would ordinarily assume. I am pointing this out early in this paper to prevent mass confusion as I progress. Later on, I will switch to a hopefully more understandable way of showing the frame ans its components.

Field Definitions

The frame is made up of several parts, called fields. Each of these fields is made up of an integral number of octets (or bytes), and serves a specific function.

Flag Field

Since amateur packet radio is a bit-oriented protocol, the only way to tell when one frame is over and another is starting for sure is to delimit each frame with a certain bit sequence both at the beginning and the end. This is the job of the flag field. A flag consists of a zero followed by six ones followed by another zero, or 01111110 (7E hex). Due to the bit stuffing mentioned above, the only time this sequence is allowed is at the beginning and end of a legitimate frame.

Address Field

The address field is used to identify both where the frame came from and what the destination of it is. In the CCITT recommendation X.25, this field is only one octet long. This permits at most 256 users per level 2 channel, and since some bits of this field were used for other purposes, the real number of users were about thirty per level 2 channel. Both the HDLC and ADCCP recommendations allowed the address field to be extended, so we decided to extend the address field per their recommendations in the amateur version of X.25 to include the callsigns of both the destination and source amateur radio stations.

The method used to extend the address field will be described shortly.

Control Field

The control field is used to identify the type of frame and control several attributes of the level 2 connection. It is one octet in length, and its encoding will be discussed in a following section.

PID-Field

The Protocol Identifier (PID) field is used only in information frames, and identifies what

kind of layer 3 protocol, if any, is in use. Its encoding is as follows:

M S B	L S B	
xx00xxxx		Reserved at the moment.
xx01yyyy		AX.25 layer 3 implemented.
xx10yyyy		AX.25 layer 3 implemented.
11110000		No layer 3 implemented.
11111111		Escape character. Next byte contains more PID information.

Where:

1. An x indicates a "don't care" bit.
2. A y indicates all combinations used.

Information Field

The information field is used to convey the actual user data from one end of the link to the other. I fields are allowed in only three tyes of frame;, the I frame, the UI frame, and the FRMR frame. The I field can be up to 256 octets long, and should be an even multiple of octets long. Any information in the I field should be passed along the link totally transparently, except for any zero-bit insertion necessary to prevent flags from accidentally appearing in the I field.

Frame Check Sequence

The frame-check sequence is a sixteen-bit number calculated by both the sender and receiver of a frame. It is used to make sure that the frame was not corrupted by the medium used to get the frame from the sender to the receiver. It is calculated in accordance with ISO 3309 (HDLC) recommendations.

Bit Stuffing

In order to assure that the flag sequence mentioned above doesn't accidentially appear anywhere else in a frame, as the frame is being sent it should be monitored, and if more than five contiguous ones are detected, a zero bit should be added between the fifth and sixth ones, eliminating the possibility of a flag appearing in the frame other than where it belongs. The receiver of five ones, a zero, and more ones should automatically eliminate the inserted zero before passing the data on.

Bit Order of Transmission

With the exception of the FCY field, all other fields in an AX.25 frame should be sent starting with the least-significant bit. In accordance with HDLC practices, the FCS should be sent most-significant bit first.

Frame Abort

If a frame must be prematurely aborted, at least fifteen contiguous ones should be sent with no bit stuffing added.

Invalid Frames

Any frame consisting of less than 136 bits, or not bounded by opening and closing flags, or not octet aligned (an integral n-umber of octets) should be considered an invalid frame by the link layer.

Address Field Encoding

The address field of all frames should be encoded with both the destination and source amateur callsigns of the frame. If a level 2 amateur "repeater" is to be used, its callsign should also be in the address field. AX.25 follows the HDLC recommended method of extending the address field in order to fit all this information into the address field.

Basically, the way the HDLC address field is extended beyond one octet is to reserve the least-significant bit of each octet for what is called an "extender bit". This bit is set to zero if the next octet contains more address field information, and is set to one if this is the last octet. To make room for this extender bit, the amateur radio call sign information is shifted one bit to the left.

The actual encoding techniques for both non-repeater and repeater operation follows.

Non-Repeater Address--Field Encoding

If a level 2 repeater is not being used, the address field is encoded as shown in Fig. 2. The destination address is the call sign of the amateur radio station that the frame is addressed to, while the source address contains the amateur call sign who sent the frame. These call signs are the call signs of the two ends of a level 2 AX.25 link only, not of any other station, such as the destination of a packet going thru an intermediary link. Those addresses should be in a higher layer, not layer 2.

A1 thru A14 are the fourteen octets that make up the two address sub-fields of the address field. The destination sub-address is seven octets long (A1 thru A7), and is sent first. This will allow the receivers of the frame time to check the destination address sub-field and see if the frame is for them while the rest of the frame is being received. The source address sub-field is then sent in octets A8 thru A14. Both of these sub-fields are encoded in the same manner, except for the last octet having the HDLC address extender bit set. Since they are basically the same, only the destination sub-address will be outlined.

There is an extra octet at the end of each address sub-field that allows room for a Secondary-Station Identifier (SSID) and three additional bits for future expansion. The SSID field allows an Amateur Radio operator to have more than one packet radio station. This is useful when an amateur wants to put up a repeater in addition to his regular station for example.

Appendix A shows a typical AX.25 frame in the non-repeater mode.

Destination Sub-Field, Encoding

Fig. 3 shows how an amateur call sign is placed in the destination address sub-field, occupying octets A1 thru A7.

Octet	ASCII	Bin.Data	Hex Data
A1	W	10101110	AE
A2	B	10000100	84
A3	4	01101000	68
A4	J	10010100	94
A5	F	10001100	8C
A6	I	10010010	92
A7	SSID	0RRSSID0	

Bit Position--> 76543210

Fig. 3. Destination Field Encoding

Where:

1. The top octet (A1) is the first octet sent (sort of like popping it off the top of the stack), with bit 0 of each octet being the first bit sent, and bit 7 being the last bit sent.

2. The first (low order or bit 0) bit of each octet is the HDLC extender bit, which is set to zero on all but the last octet in the address field, where it is set to one.

3. The bits marked "R" are reserved bits. they may be used in an agreed upon manner in individual networks. If they aren t implemented, they should be set to one.

4. The characters of the callsign should be standard seven-bit ASCII (upper case only) before being shifted left to make room for the extender bit. If the callsign is less than six characters long it should be padded at the trailing end with ASCII spaces between the end of the callsign and the SSID octet.

5. The SSID portion of the last octet has been intentionally left vague at this point, and is left up to the individual station to assign. The only recommended restriction is to reserve the all-one condition (1111) for an all-call SSID in case one wants to reach an amateur but dooesn't know what SSID that amateur operates under.

Level 2 Repeater Address-Field Encoding

If a frame is to go thru a level 2 amateur packet repeater, there is an additional address sub-field added to the end of the address field. This additional sub-field contains the call sign of the repeater to be used. This will allow more than one repeater to share the same rf channel, which has been a problem with the older protocols. If this field exists, the last octet of the source sub-field has its extender bit set to zero, indicating that more address-field data follows. The repeater address sub-field is encoded in the same manner as the destination and source address sub-fields, except for one bit in the last octet, called the "H" bit. The H bit is used to indicate whether a frame has been repeated or not. This is necessary to prevent someone from potentially receiving two identical frames, the one going to the repeater, and the one coming back from the repeater. Fig. 4 shows how the repeater address sub-field is encoded. Appendix B is an example of a complete frame on its way back from a repeater.

Octet	ASCII	Bin.Data	Hex Data
A15	W	10101110	AE
A16	B	[illegible]	
A17	4	01101010	10000684
A18	J	10010100	94
A19	F	10010010	10001192
A20	I	[illegible]	
A21	SSID	HRRSSID1	

Bit Order --> 76543210

Fig 4. Repeater Address Encoding

Where:

1. The top octet is the first octet sent, with bit 0 being sent first, bit 7 sent last of each octet.

2. As with the source and destination address sub-fields discussed above, bit 0 of each octet is the HDLC address extender bit, which is set to zero on all but the last address octet (A21) where it is set to one.

3. The "R" bits are reserved just like in the source and destination sub-fields.

4. The "H" bit is the has-been-repeated bit. It is set to zero on a non-repeated frame, and set to one by the repeater when the frame has been repeated.

It should be noted that some of the advantages of this addressing scheme are mentioned in Appendix C.

Control Field Formats

The control field is responsible for identifying what type of frame is being sent, and is also used to convey commands and responses from one end of the link to the other to maintain proper control over the link.

The control fields used in AX.25 use the CCITT X.25 control fields for balanced operation, with an additional control field taken from ADCCP to allow connectionless and round-table operation.

There are three general types of AX.25 frames. They are the I formation frame (I frame), the Supervisory frame (S frame), and the Unnumbered frame (U frame). Fig. 5 shows the basic format of the control field associated with these types of frames.

Control Field Type	Control Field Bits 7 6 5	4	3 2	1	0
I Frame	N(R)	/P/E	N(S)		0
S Frame	N(H)	P/F	s S	0	1
U Frame	M M M	P/r'	M M	1	1

Fig. __ Control Field Formats

Where:

1. Bit 0 is the first bit sent, bit 7 is the last bit sent of the control field.

2. N(S) is the send sequence number (bit 2 is the LSB).

3. N(R) is the receive sequence number (bit 6 is the LSB).

4. The "S" bits are the supervisory function bits, and their encoding is discussed below.

5. The "M" bits are the unnumbered frame modifier bits and their encoding is discussed below.

6. The P/F bit is the Poll/Final bit. Its function is described in more detail shortly.

Control Field Definitions

Information Frame Control Field

All I frames have bit 0 of the control field set to zero. N(S) is the sender's send sequence number (the send sequence number of this frame). N(R) is the sender's receive sequence number (the sequence number of the next expected received frame. These numbers are described in the section regarding flow control.

Supervisory Frame Control Field

Supervisory frames are denoted by having bit 0 of the control field set to one, and bit 1 of the control field set to zero. S frames provide supervisory link control such as acknowledging or requesting retransmission of I frames, and link level window control. Since S frames don't have an information field, the sender's send variable and the receiver's receive variable are not incremented for S frames.

Unnumbered Frame Control Field

Unnumbered frames are distinguished by having both bits 0 and 1 set to one. U frames are responsible for maintaining control over the link beyond what is accomplished with S frames. They are also responsible for the establishment and tearing down of the link. U frames also allow for the transmission and reception of information outside of the normal flow control. Some U frames may contain information fields.

Control Field Parameters

Sequence Numbers and Variables

Every AX.25 I frame shall be assigned a sequential number from 0 to 7. This will allow up to seven outstanding I frames per level 2 connection at a time.

Send State Variable V(S)

The send state variable is an internal variable that is never sent. It contains the next sequential number to be assigned to the next transmitted I frame. This variable is updated upon the transmission of each I frame.

Send Sequence Number N(S)

The send sequence number is found in the control field of all I frames. It contains the sequence number of the I frame being sent. Just prior to the transmission of the I frame, N(S) is updated to equal the send state state variable.

Receive State Variable V(R)

The receive state variable is an internal variable that contains the sequence number of the next expected received I frame. This variable is updated upon the reception of an error-free I frame whose send sequence number equals the present received state variable value.

Received Sequence Number N(R)

The received sequence number is in both I and S frames. Prior to sending an I or S frame,

this variable is updated to equal that of the received state variable, thus implictly acknowledging the proper reception of all I frames up to and including N(R)-1.

Poll/Final (P/F) Bit

The P/F bit may be used in all types of frames. It is used in a command (poll) mode to request an immediate reply to a frame. The reply to this poll is indicated by setting the response (final) bit in the appropriate frame. Only one outstanding poll condition per direction is allowed at a time.

Control Field Encoding

Information Frame Control Field

The information frame control field is encoded as shown in Fig. 6. These frames are sequentially numbered to maintain control of their passage over the link level connection.

Control Field Bits			
7 6 5	4	3 2 1	0
N(R)	P/F	N(S)	0

Fig. 6. I Frame Control Field

Supervisory Frame Control Field

The supervisory frame control fields are encoded as shown in Fig. 7. In AX.25, S frames are used only as responses to other frames.

Control Field Bits		7 6 5	4	3 2	1 0
Receive Ready	RR	N(R)	P/F	0 0	0 1
Receive Not Ready	RNR	N(R)	P/F	0 1	0 1
Reject	REJ	N(R)	P/F	1 0	0 1

Fig. 7. S frame control Fields

Receive Ready (RR) Response

Receive Ready is used to do the following:

1. To indicate that the sender of the RR is now able to recieve more I frames.

2. To acknowledge properly received 1 frames up to, and including N(R)-1.

3. To clear a previously set busy condition created by an RNR command having been sent.

It should be noted that the status of the other side of the link can be requested by setting the poll bit.

Receive Not Ready (RNR) Response

Receive not ready is used to indicate to the sender of I frames that receiver is temporarily busy and cannot accept any more I frames. Frames up to N(R)-1 are acknowledged. Any I frames numbered N(R) and higher that might have been caught in between and not acknowledged when the RNR command was sent are NOT acknowledged.

The RNR condition can be cleared by the sending of a UA, RR, REJ or SABM frame. The P/F bit can be used within the RNR frame to interrogate the status of the other side of the link.

Reject (REJ) Response

The reject frame is used to request retransmission of I frames starting with N(R). Any frames that were sent with a sequence number of N(R)-1 or less are acknowledged. Additional I frames may be appended to the retransmission of the N(R) frame if there are any.

Only one reject frame condition is allowed in each direction at a tine. The reject condition is cleared by the proper reception of I frames up to the I frame that caused the reject condition to be initiated.

As with the other supervisory responses, the P/F bit may be used in the REJ frame.

Unnumbered Type Frames

Unnumbered frame control fields are either commands or responses. This standard follows X.25 as much as possible. The only deviation from X.25 is in the addition of the Unnumbered Information (UI) frame from ADCCP. X.25 is designed to work with in full-duplex systems with only one main device (DCE) and potentially many users (DTEs).

Amateur Radio packet systems differ greatly on both of these respects. Not only is Amateur Radio packet networking done in a half-duplex rf environment, but many DCE/DTE links many be sharing the same channel. Many amateurs have rejected the use of X.25 as a result of these problems. X.25 can easily be enhanced so that it will perform properly over amateur radio.

Fig. 8 shows the layout of U frames implement ed within this standard.

Control Field	Type	Control Field Bits 7 6 5	4	3 2	1 0
Set Asynchronous Balanced Mode-SABM	Cmd	0 0 1	P	1 1	1 1
Disconnect-DISC	Cmd	0 1 0	P	0 0	1 1
Disconnected Mode DM	Res	0 0 0	P/F	1 1	1 1
Unnumbered Acknowledge-UA	Res	0 1 1	F	0 0	1 1
Frame Reject-FRMR	Res	1 0 0	F	0 1	1 1
Unnumbered Information-UI	Either	0 0 0	P/F	0 0	1 1

Fig. 8. U Frame Control Fields

Set Asynchronous Balanced Mode (SABM) Command

The SABM command is used to place 2 stations in the asynchronous balanced mode. This is a balanced mode of operation known as LAP B where DCEs and DTEs are treated as equals.

Information fields aren't allowed in SABM commands. Any outstanding I frames left when the SABM command is issued will remain unacknowledged.

Disconnect (DISC) Command

The DISC command is used to terminate a link session between two stations. No information field is permitted in a DISC command frame. Any outstanding I frames will remain outstanding.

Disconnected Mode (DM) Response

The disconnected mode response is sent whenever the DTE or DCE receives a frame other than a SABM while in a disconnected mode. It is sent to request a set mode command, or to indicate it cannot accept a connection at the moment. The DM response cannot have an information field.

A DCE or DTE in the disconnected mode will respond to any command other than a SABM witn a DM response with the P/F bit set to 1.

Unnumbered Acknowledge (UA) Response

The UA response frame is sent to acknowledge the reception and acceptance of a U frame command. A received command is not actually processed until the UA response frame is sent. An information field is not permitted in a UA frame.

Frame Reject (FRMR) Response

The FRMR response frame is sent to report that for some reason the receiver of a command or information frame cannot successfully process that frame and that the error condition is not correctable by sending the offending frame again. Typically this condition will appear when a frame without an FCS error has been received with one of the following conditions:

1. The reception of an invalid or not

implemented command or response frame.

2. The reception of an I frame whose information field exceeds the agreed upon length.

3. The reception of an improper N(R). This usually happens when the N(R) frame has already been sent and acknowledged, or when N(R) is out of sequence with what was expected.

4. The reception of a frame with an information field where one is not allowed or the reception of an U or S frame whose length is incorrect.

When a CMDR or FRMR frame is sent, an information field is added to the frame that helps to explain where the problem occurred. This information field is three octets long and its contents is shown if Fig. 9 below.

```
--------------------------------------------------
|                Information Field Bits           |
| 2 2 2 2 1 1 1 1 1 1 1 1 1 1                      |
| 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0  |
|-------------------------------------------------|
| 0 0 0 0|Z|Y|X|W| V(R)|C| V(S)|0| Rejected Frame |
|        | | | | |     | |     | | Control Field  |
--------------------------------------------------
```

Fig. 9. FRMR Frame Information Field

Where:

1. The rejected frame control field carries the control field of the frame that caused the reject condition. It is in bits 1-8 of the information field.

2. V(S) is the current send state variable f the device reporting the rejection 4bit 10 is the low bit).

3. V(R) is the current receive state variable of the device reporting rejection (bit 14 is the low bit).

4. If W is set to 1, the control field received was invalid or not implemented.

5. If X is set to 1, the frame that caused the reject condition was considered invalid because it was a U or S frame that had an information field that is not allowed. Bit W must be set to 1 in addition to the X bit.

6. If Y is set to 1, the information field of a received frame exceeded the maximum capacity of the device reporting the condition.

7. If Z is set to 1, the control field received and returned nibits 1 to 8 contained an invalid N(R).

8. Bits 8, and 20 to 23 are set to 0. Bit 12 is set to 0 if the rejected frame was a command, or 1 if if it was a response.

Unnumbered Information (UI) Frame

The unnumbered information frame is used to pass information along the link outside the normal information controls. This allows information fields to go back and forth on the link bypassing flow control. Since these frames are NOT acknowledgeable, if one gets wiped out, there is no way to recover it.

The UI frame is not defined in X.25. It has been taken from ADCCP to allow uncontrolled information to flow thru the link without interfering with a next higher layer.

Link Error Recovery

There are several link-level errors that are recoverable without tearing down the connection. These error situations may occur as a result of malfunctions within the DTE or DCE, or if transmission errors occur.

Invalid Frame or FCS Error

If an invalid frame is received or a frame is received with an FCS error, that frame will be discarded with no action taken.

Device Busy Condition

When a DTE or DCE becomes temporarily busy, such as when receive buffers are full, it will send a receive not ready (RNR) frame. This tells the other side of the link that the device cannot handle any more I frames at the moment. This condition is usually cleared by the sending of a UA, RR, REJ, or SABM command frame.

Send Sequence Number Error

If the send sequence number, N(S), of an otherwise error-free received I frame does not match the receive state variable, V(R), a send sequence error has occured, and the information field will be discarded. The receiver will not acknowledge this frame, or any other I frames until N(S) matches V(R).

The control field of the erroneous I frame(s) will be accepted so that link supervisory functions can still be performed, such as checking the P/F bit. Because of this updating, the retransmitted I frame may have an updated P bit and N(R).

Reject (REJ) Error Recovery

REJ is used to request a retransmission of I frames following the detection of a sequence error. Only one outstanding reject condition is allowed at a time. This condition is cleared when the requested I frame has been received.

A device receiving the REJ command will clear the error by sending over the I frame indicated in H(R) of of the HEJ command frame.

Time-out Error Recovery

When a transmission abnormality wipes outa single I frame, or the last I frame of a group, there is no way of telling this immediately, since the receiver does not necessarily know something was sent until another frame is sent resulting in an out-of-sequence error. To cope with this situation better, some form of time-out delay will be incorporated by the sender after it sends out a frame. This time-out timer is started at the time a frame is sent, and stopped by the reception of an acknowledgement for the sent frame. If the timer times out before an acknowledgement is received, any unacknowledged frames are retransmitted. The delay is an agreed-upon amount that will vary with the type of rf medium and signaling speed used.

dejection Error

A rejection error condition occurs when an error-free received frame has one of the following problems:

1. An invalid command or response control field.

2. An invalid frame format.

3. An Invalid N(R).

4. An information field that exceeds the maximum the device can accept.

Once a rejection error occurs, no more I frames are acccepted (with the exception of the P/F bit still usable) until the error is resolved. The error condition is reported to the other side of tne link by sending a FRMR response frame.

Primary/Secondary versus Balanced Operation

There are two basic classes of link-level connections. The first, known as Link Access Procedure (or LAP) is often called an unbalanced service where the DCE is considered the primary (or master) devices and the DTEs are considered secondary (or slave) devices. The second class of service is known as LAPB, Link Access Procedure Balanced. In this service both devices are treated as equals as far as connection requests and other types of commands. There is still only one DCE and potentially many DTEs, but both ends can command the link equally.

Primary/Secondary (LAP) Operation

LAP is the older style of link control,

where most of the intelligence was assumed to be in a large mainframe (the DCE) and the end users were just using smart terminals (the DTEs). Since network software can have a lot of overhead it made sense at the time to put most of the overhead in the big computer, and just enough smarts to make the link work in the terminals.

Balanced (LAPB) Operation

LAPB is a slightly modified version of LAP. It has been changed to allow the two sides of a link to operate in a more balanced manner. In the official version of X.25 there is still only one DCE to potentially many DTEs, but the two can-operate more-as equals than master and slave.

LAPB is what this document describes for use over Amateur Radio packet networks. Even when there is a network controller overseeing the network operation, the balanced link procedure will enhance operation.

Connection Operation

In amateur radio network operations, it would be very helpful if one level 2 protocol would work with the various rf systems in use. An example of this is the difference in operation between a simple two-station link, and multiple stations operating thru a network controller. Obviously, when a network controller exists, it should b considered the DCE, while the other station: connecting to it would be the DTEs. A simple two-station connection is another matter. To this type of connection the station requesting a connection should always be considered the DTE, while the device that is receiving the connection request should operate as the DCE. This simple rule should eliminate any ambiguity that might otherwise occur under these conditions.

NOTE There are a couple minor changes from the official X.25 standard in the protocol recommended here. These changes are done only as absolutely necessary to work over theshared rf media. Since X.25 was written to work so that one DCE talked with many DTEs over a closed network, it cannot properly cope with a channel where there may be many DCEs linked to many DTEs. Some amateurs have thrown X.25 out because of this problem. It seems to take just a couple minor changes in the initial link set-up procedure to make X.25 work properly over 'amateur radio. Where these changes are made, both the original X.25 procedure and the recommended amateur procedure will be noted.

LAPB Procedures

The following describes the procedures used to set-up, use, and disconnect a balanced link between a DTE and DCE. These procedures have been taken from X.25 and conform very closely to that standard, except where it was necessary to change due to the radio enviroment.

Address Field Operation

All transmitted frames shall have address fields conforming to above-mentioned rules. All frames should have both the destination device and the source device addresses in the address field, with the destination address coming first. This will allow many links to share-the same rf channel. The destination address is always the address' of the station(s) to receive the frame, while the source address contains the address of the dveice that sent the frame. The destination address can be a group name or club call however, if point to multi-point operation is allowed. This will be discussed further under link operations.

LAPB Connection Establishment

When a device (either a DCE or DTE) wishes to connect to another device, it will send a SABM command frame to that device and start a time-out timer(T1). If the other device is there and able to connect, it will answer with a UA response frame and at the same time reset both of it's internal state variables (V(S) and V(R)). The reception of the UA response frame at the other end will cause the device requesting the connection to abort the T1 timer and set its internal state varia'bles to O also.

If the other device doesn't respond before T1 times out, the device requesting the connection will re-send the SABM frame, and start T1 running again. This trying to establish- a connection will continue until the requesting device has tried unsuccessfully a number of times. That number (N1) is variable, depending on the frequency of operation, type of transmission (eg. terrestial vs. satellite), and the signaling speed in use. N1 will be discussed in another section.

Information Transfer

Once a connection has been established as outlined above, both devices are able to accept I, S, and U frames.

Sending of I Frames

Whenever a station has an I frame to transmit, it will send the I frame with N(S) of the control field equal to it's current send state variable V(S). Once the I frame is sent, the send state variable is incremented by one.

The station should not transmit any more I frames if it's send state variable equals the last received N(R) from the other side of the link plus seven. If it were to send more I frames, the flow control window would be exceeded and errors could result.

If a device is in a busy condition, it may still send I frames as long as the other device is not also busy.

If a device is in the frame-rejection mode, it will stop sending I frames.

Receiving I Frames

If a device receives a valid I frame (one with a correct FCS and whose send sequence number equals the receiver's receive state variable) and is not in the busy condition, it will accept the received I frame, increment it's receive state variable, and act in one of the following manner:

1. If it has an I frame to send, that I frame may be sent with the transmitted N(R) equal to it's receive state variable V(R) (thus acknowledging the received frame. Alternately, the device may send an RR frame with N(R) equal to V(R), and then send the I frame.

2. If there are no outstanding I frames, the receiving device will send an RR frame with N(R) equal to V(R).

If the device is in a busy condition it may ignore any received I frames without reporting this condition other than repeating the indication of the busy condition.

If a busy condition exists, the station receiving the busy condition indication should poll the sender of the busy indication periodical-l-y until the busy condition disappears.

The reception of I frames that contain zero length information fields shall be reported to the next level but no information field will be transferred.

When an I frame is received with a correct FCS but it's send sequence number does not match the current receiver's receive state variable, the frame should be discarded and a HEJ frame should be sent with a receive sequence number equal to one higher (modulo 8) than the last correctly received I frame . Any out-of-sequence received I frames should be handled in this manner. The received state variable and poll bit in such a discarded frame should be checked before throwing it away, and take any action needed depending on the condition of them.

Receiving Acknowledgement

Whenever an I or S frame is correctly received, even in a busy condition, the N(R) of the received frame should be checked to see if it includes an acknowldegment of outstanding sent I frames. The T1 timer should be reset if the received frame actually acknowledges previously unacknowledged frames. If the T1 timer is reset, and there are still some frames that have been sent that are not acknowledged, T1 should be

started again. If the T1 timer tins out before an acknowledgement is received, the device should proceed to the retransmission procedure.

Receiving Reject

Upon receiving a REJ frame, the transmitting station will set its send state variable to the same value are the REJ frames received sequence number in the control field. The device will then retransmit any I frame(s) outstanding at the next available opportunity conforming to the following:

1. If the device is not transmitting at the time, and the channel is open, the device may commence to retransmit the I frame(s) immediately.

2. If tne device is operating on a full duplex channel transmittiong a U or S frame when it receives a REJ frame, it may finish sending the U or S frame and then retransmit the I frame(s).

3. If the device is operating in a full duplex channel transmitting another I frame when it receives a REJ frame, it may abort the I frame it was sending and start retransmission of the requested I frames immediately.

4. The device may send just the one I frame outstandin g or it may send more than one if any more I frames followed the first one not acknowledged, provided the total to be sent does not exceed the flow control window (7 frames).

If tne device recives a REJ frame with the poll bit set, it should respond with either an RR or RNR frame with the final bit set before retransmitting the outstanding I frame(s).

Receiving an RNR Frame

Whenever a device receives an RNR frame, it may transmit or retransmit the I frame whose send sequence number equals that of tne received sequence number indicated in the RNR control field. If timer T1 runs out after the RNR was received the waiting acknowledgement procedure listed below should be performed. The poll bit may be used in conjunction with S frames to test for a change in the condition of the busied out station. No I frames other than the one mentioned above may be sent out before the busy condition is cleared.

Sending a Busy Indication

Whenever a device enters a busy condition, it will indicate this by sending an RNR response at the next opportunity. While the device is in the busy condition, it may receive and process S frames. and if a received S frame has the P bit set to one, the device should send a RNR frame with the F bit set to one at the next possible opportunity. To clear the busy condition, the device should send either a RR or REJ frame with the received sequence number equal to the current receive state variable, depending on whether the last received I frame was properly received or not.

Waiting Acknowledgement

The device should maintain an internal retransmission count variable which is set to zero whenever another I frame is acknowledged (either thru the reception of a UA or RNR frame, or when a received I or S frame has an N(R) higher than the last received N(R), showing the acknowledgement of additional I frames).

Any time the timer T1 runs out, the device will re-enter the timer recovery condition, the retransmission count variable will be incre- ment ed by one, and another internal variable (X) will be set to the current send state variable value.

The device will then restart the T1 timer, set its receive state variable to the last receive sequence number, and retransmit the corresponding I or S frame with the P bit set to one.

The timer recovery condition is cleared when the device receives a valid S frame with the F bit set to one.

If the device receives an S frame with the F bit set to one and N(R) within the range from the current send state variable to X mentioned above inclusive while in the timer recovery condition, this condition will be cleared, and the send state variable will be set to the N(R) received.

If the device. receives an S frame with the F bit set to zero but otherwise the same condition as the last paragraph, the timer recovery condition will NOT be cleared. The received N(R) may be used however to update the send state variable. The device may keep the last I frame transmitted (even if it was acknowledged) to be retransmitted with the P bit set to one if timer T1 expires at a later time.

Once the retransmission count variable reaches N2, the device should proceed to the resetting procedures outlined below.

Link Disconnection

When in the information-transfer phase, either device may initiate a link disconnection by sending a DISC frame. It should then start its T1 timer, and wait for a response. If the proper response doesn't come before T1 times out, it should send the DISC frame again and restart T1. If this happens N2 times, the device should enter the disconnected state.

When a DISC frame is received, the receiver should return a UA response frame,, and enter the disconnected state.

Disconnected State

After having sent a DISC frame and received a UA, or receiving a DISC and having sent a UA, the device will enter the disconnected state.

In the disconnected state, the device may initiate a link set-up as outlined in connection establishment above. It may also respond to the reception of a SABM and establish a connection, or it may ignore the SABM and send a DM instead.

Any station receiving a DISC command while in the disconnected state should send back a DM response frame.

Any device receiving a command frame other than a SABM or UI frame with the P bit set to one should respond with a DM frame with the F bit set to one. The offending frame should also be ignored.

When the device enters the disconnected state after an error condition or if it has recovered from an internal error condition by coming up in the disconnected state, it should indicate this by sending a DM response rather than a DISC frame. It should start the T1 timer when the DM is sent, and if T1 times out before getting a SABM or DISC frame back, it should send another DM frame and restart T1. After retransmitting the DM frame N2 times the device will remain in the disconnected state, and no other action will be taken.

Resetting Procedure

The resetting procedure is used to initialize both directions of flow after a nonrecoverable error has occured. This resetting procedure is only used when in the information transfer phase of an AX.25 link.

A device shall request a reset by sending an SABM frame. Upon receiving an SABM frame from a station previously connected to, the receiver of an SABM frame should send a UA frame back at the earliest opportunity. Both devices should then set their send and receive state variables to zero. Any busy condition that previously existed will also be cleared.

It is possible to initiate a disconnect procedure instead of resetting the link.

One device may ask the other to reset the link by sending a DM response frame. After the DM frame is sent, the sending device will then enter the disconnected state.

One device may ask the other to initiate a link reset by transmitting a FRMR response frame.

After sending the FRMR frame, The sending device will enter the frame re 'ect state. This condition is cleared when the device that sent the FRMR frame receives an SABM or DISC command, or a DM response frame. Any other command received while the device is in the frame reject state will cause another FRMR to be sent out with the same information field as originally sent.

The device that sent the FRMR fram should start the T1 timer when the FRMR is sent If above mentioned frames are not received before the timer runs out the FRMR frame should be retransmitted, and the T1 timer restarted as described in the waiting acknowledgement section above. If the FRMR is sent N2 times without success, the link should be reset.

Rejection Conditions

A device should initiate the link-reset procedure when a frame is received with the correct FCS and address field during the information transfer phase withone or more of the following conditions:

1. The frame is not known as a command or response to the device.
2. The information field is invalid (as an example is longer than 256 octets).

A device will initiate a reset procedure whenever it receives a DM or FRMR response frame during the information transfer phase.

A device may initiate a reset procedure also whenever it receives a UA response frame or if it receives an unsolicited response frame with the F bit set to one.

Collision Recovery

Collisions in a Half-Duplex Enviroment

Collisions of frames of any type in a half-duplex enviroment are essentially taken care of by the retry nature of the T1 timer and retransmission count variable. No other special action needs to be taken.

Collisions in a Full-Duplex Enviroment

Collisions in a full-duplex enviroment are not really frame collisions, but have more to do with the devices being pulled in two different directions at the same time.

Collisions of Unnumbered Commands

If the sent and received U command frames are the same, both devices should send a UA response at the earliest opportunity, and both devices should enter the indicates state.

If the sent and received U commands are different, both devices should enter the disconnected state, and transmit a DM frame at the earliest opportunity.

Collision of a DM with a SABM or DISC

When an unsolicited DM response frame is sent, a collision between it and a SABM or DISC may occur. In order to prevent this DM from being misinterpreted. all unsolicited DM frames should be transmitted with the F bit set to zero. All SABM and DISC frames should be sent with the P bit set to one, so there isn't any confusion when a DM frame is received.

Connectionless Operation

In Amateur Radio circles, there is a type of operation that isn't really feasable using level 2 connections. This operation is the roundtable, where several amateurs may be engaged in one conversation. The only way to accomplish this in a connected mode would be to have everyone cross-connected with each other. This would require a seperate frame to be sent to each member of the roundtable every time someone says something. Obviously, this mode is not practical. The way most amateur packet radio enthusiasts have ended up implementing the roundtable operation is outside the AX.25 connection, but still using the AX.25 frame structure. AX.25 does allow a special frame for this operation, called the Unnumbered Information (UI) frame. It is recommended that when this type of operation is in use, the destination address have a code word installed in it to prevent the users of that particular roundtable from seeing all frames going thru the shared RF medium. An example of this is if a group of amateurs are in a roundtable discussion about packet radio, they could put PACKET in the destination address, so they would only receive frames from others in the same discussion. An added advantage of the use of AX.25 in this manner is that the source of each frame is in the source address sub-field, so software could be written to automatically display who is making what comments.

Admittedly, this is a kludge to the level 2 AX.25 protocol. THis type of operation really belongs at the next layer (layer 3, packet level) of operation, but until layer 3 is implemented, this appears to be an acceptable substitute.

Keep in mind that this mode is connectionless, so all transmitted frames should be of good quality, as there will be no requests for retransmissions of bad frames. Collisions will also occur, with the potential of losing the frames that collided.

List of System Defined Parameters

Timers

It is recommended that there are two timers used to maintain the integrity of the AX.25 layer 2 connection.

The first timer, T1, is used to make sure a device doesn't wait forever for a response to a frame it sends. This timer cannot be expressed in absolute time, since the time required to send frames varies greatly with the baud rate used at level 1. T1 should be at least twice the time it would take to send a maximum length frame to the other end of the link, and get the proper response frame back from the other end of the link. This would allow time for the other end of the link to do some processing before responding.

The second timer, T2, is used whenever T1 isn't running to make sure that a supervisory frame is sent periodically to maintain link integrity. It also will vary dramatically depending on layer 1 constraints, and is subject for further study.

Maximum Number of Retrys (N2)

The maximum number of retrys is used in conjunction with the T1timer. It will vary depending on the layer 1 in use, but will generally be sixteen.

Maximum Number of Octets in an I Field (N1)

The maximum number of octets allowed in the I field will be 256. There should also be an even multiple number of octets.

Maximum Number of I Frames Outstanding (k)

The maximum number of outstanding I frames at a time is seven. A smaller number may be used at any time, provided it is agreed upon ahead of time.

First
Bit Sent

Flag	Address	Control	FCS	Flag
Ø111111Ø	112/168 Bits	8 Bits	16 Bits	~01111110

Fig. 1A. U and S Frame Construction

First
Bit Sent

Flag	Address	Control	PID	Info.	FCS	Flag
01111110	112/168 Bits	8 Bits	8 Bits	N*8 Bits	16 Bits	01111110

Fig. 1B. Information Frame Construction

First
Octet Sent

<table>
<tr><td colspan="14">Address Field of Frame</td></tr>
<tr><td colspan="7">Destination Address</td><td colspan="7">Source Address</td></tr>
<tr><td>A1</td><td>A2</td><td>A3</td><td>A4</td><td>A5</td><td>A6</td><td>A7</td><td>A8</td><td>A9</td><td>A10</td><td>A11</td><td>A12</td><td>A13</td><td>A14</td></tr>
</table>

Fig, 2A. Non-Repeater Address Field Encoding

First
Octet Sent

<table>
<tr><td colspan="21">Address Field of Frame</td></tr>
<tr><td colspan="7">Destination Address</td><td colspan="7">Source Address</td><td colspan="7">Repeater Address</td></tr>
<tr><td>A1</td><td>A2</td><td>A3</td><td>A4</td><td>A5</td><td>A6</td><td>A7</td><td>A8</td><td>A9</td><td>A10</td><td>A11</td><td>A12</td><td>A13</td><td>A14</td><td>A15</td><td>A16</td><td>A17</td><td>A18</td><td>A19</td><td>A20</td><td>A21</td></tr>
</table>

Fig. 2B. Repeater Address Field Encoding

Appendix A. Non-Repeater Frame Example

Octet	ASCII	Bin.Data	Hex Data
Flag		01111110	7E
A1	K	10010110	96
A2	8	01110000	70
A3	M	10011010	9A
A4	M	10011010	9A
A5	0	10011110	9E
A6	space	01000000	40
A7	SSID	01100000	60
A8	W	10101110	AE
A9	B	10000100	84
A10	4	01101000	68
A11	J	10010100	94
A12	F	10001100	8C
A13	I	10010010	92
A14	SSID	01100001	61
Control	SABM	**00111111**	3F
PID	none	**11110000**	F0
PCS	part 1	XXXXXXXX	HH
FCS	part 2	XXXXXXXX	HH
Flag		01111110	7E

The frame shown is an SABM frame, *not going thru* a level 2 repeater, from WB4JFI (SSID=0) to K8MMO (SSID=0), with no level 3 protocol.

Appendix B. Repeater Type Operation

Octet	ASCII	Bin.Data	Hex Data
Flag		01111110	7E
A1	K	10010110	96
A2	8	01110000	70
A3	M	10011010	9A
A4	M	10011010	9A
A5	0	**10011110**	9E
A6	space	01000000	40
A7	SSID	01100000	60
A8	W	10101110	AE
A9	B	10000100	84
A10	4	01101000	68
A11	J	**10010100**	94
A12	F	10001100	8C
A13	I	10010010	92
A14	SSID	01100000	60
A15	W	**10101110**	AE
A16	B	10000100	84
A17	4	01101000	68
A18	J	10010100	94
A19	F	10001100	8C
A20	I	10010010	92
A21	SSID	11100011	E3
Control	SABM	00111111	3F
PID	none	11110000	F0
FCS	part 1	XXXXXXXX	HH
FCS	part 2	XXXXXXXX	HH
Flag		01111110	7E

The above frame is the same as in Appendix A, but with the addition of a repeater address sub-field (WB4JFI, SSID=1). The H bit is set, indicating this is from the output of the repeater.

Appendix C.

Advantages of the WB4JFI Addressing Scheme

Some of the advantages to using this addressing system are:

1. Every packet station will have a unique fixed address that doesn't change every time a new network is logged into.
2. Relocating to a new area won't cause major (or minor) problems.
3. Allows for more than 62 or 31 users at a time.
4. No local packet guru is needed to assign addresses with attendant concerns of backup and transfer during failure.
5. Direct or network operation requires no change of address.
6. All the problems with dynamic allocation/de-allocation are eliminated.
7. Reduces local co-network interference due to users in overlapping local network rf domains with the same address fields.
8. With every frame having both the destination and source addresses in them, it will be a lot easier to set-up and run multiple connections on the same data channel without having problems arise as to who is sending what frames to whom.
9. In round-table operation, every frame sent will have the source address imbedded in it, allowing automatic printing of the source of the frame.

Appendix D. Layer 2 AX.25 State Table

State	I with Poll	I without P	RR with Final	RR without F	REJ with Final	REJ without F	RNR with Final	RNR without F	SABM either	DISC! either	UA either	DM either	FRMR either
S1 Disconnected	DM		DM		DM		DM		UA,S5	DM		SABM,S2	
S2 Link Setup									UA	DM,S1	S5	S1	
S3 Frame Reject	FRMR		FRMR		FRMR		FKMR		UA,S5	UA,S1		SABM,S2	SABM,S2
S4 Disconnect Rqst									DM	UA	Sl	S1	
S5 Information Xfr	RR	I	I	I	I	I	S9	S9	UA	UA,S1	SABM,S2	SABM,S2	SABM,S2
S6 REJ Frame Sent	RR,S5	I,S5	I	I	I	I	S15	S15	UA,S5	UA,S1	SABM,S2	SABM,S2	SABM,S2
S7 Waiting Acknow.	RR	I	I,S5	I	I,S5	I	S9	S12	UA,S5	UA,S1	SABM,S2	SABM,S2	SABM,S2
S8 Device Busy	RNR	RNR	I	I	I	I	S10	S10	UA	UA,S1	SABM,S2	SABM,S2	SABM,S2
S9 Remote Device Busy	RR	RR	I,S5	I,S5	I,S5	I,S5			UA,S5	UA,S1	SABM,S2	SABM,S2	SABM,S2
S10 Both Devices Busy	RNR	RNR	I,S8	I,S8	I,S8	I,S8			UA,S8	UA,S1	SABM,S2	SABM,S2	SABM,S2
S11 Waiting Acknow. and Device Busy	RNR	RNR	I,S8	I	I,S8	I	S10	S13	UA,S8	UA,S1	SABM,S2	SABM,S2	SABM,S2
S12 Waiting Acknow. and Remote Busy	RR	RR	I,S5	I,S7	I,S5	I,S7			UA,S5	UA,S1	SABM,S2	SABM,S2	SABM,S2
S13 Waiting Acknow. Both Devices Busy	RNR	RNR	I,S8	I,S11	I,S8	I,S11			UA,S8	UA,S1	SABM,S2	SABM,S2	SABM,S2
S14 REJ Sent and Device Busy	RNR	RNR	I	I	I	I	S16	S16	UA,S8	UA,S1	SABM,S2	SABM,S2	SABM,S2
S15 REJ Sent and Remote Busy	RR,S9	RR,S9	I,S6	I,S6	I,S6	I,S6			UA,S5	UA,S1	SABM,S2	SABM,S2	SABM,S2
S16 REJ Sent and Both Devices Busy	RNR	RNR	I,S14	I,S14	I,S14	I,S14			UA,S5	UA,S1	SABM,S2	SABM,S2	SABM,S2

Terry Fox, WB4JFI
Vice-President, AMRAD
1819 Anderson Road
Falls Church, VA 22043

Introduction

Now that the amateur packet radio community seems to have agreed on a protocol at layer 2, the link level of the Open System Interface Reference Model (OSI-RM), it appears that it is time to begin work on the layer 3, or network layer. Layer 3 is actually made up of two sub-layers, a local or metropolitan network sub-layer, and an internetwork sublayer.

The local area network is responsible for the proper transfer of packets among a group of local users. The term local can be misleading, as a local network could actually be a network operating on hf, where the participants are actually spread out over a large geographical area. A local network is generally considered a group of devices interconnected directly together at the layer immediately above the link layer. These devices may be corresponding directly, or they may be operating thru an intermediary such as a local (or metropolitan) network controller.

The internetwork sub-layer is half a step above the local network, and is used to interconnect individual local networks. This allows a user on one local network to communicate with another user on a different local network. Depending on how the internet sublayer operates another layer above it (layer 4, transport layer) may be required to assist in the re-assembly of data sent over the internet layer.

Types of Network Operation

There are two basic types of networks, both at the local and internet sublayers. One is the connection type, and the other is the datagram type.

Connection Type Networks

The connection type of network requires a connection be established and maintained between the two devices wishing to communicate before any data can be transferred. The connection looks very much like the HDLC type layer 2 links that are established before I frames may be passed along the link.

The connection network is like a small town telephone company. Whenever a local call is made, as long as it is between the same two people, the connection will be made the same way every time (usually for a different reason though, because that's the only way a connection can be made between the two people, not necessarily the best way). in a connection network, once the connection is made, all frames MUST follow the same path. If anything should happen to that path, the connection must be torn down and re-established over again.

The main advantages of the connection type of network are.:

1. Once the connection is established, very little overhead is required to maintain proper operation.

2. Generally, a connection oriented network does not allow data packets to be received out of their proper sequence, thereby greatly simplifying the transport layer required.

Some of the disadvantages of the connection type of network are:

1. Once a connection is established, all packets must follow the path generated while the connection was being made. This could be a problem if either the network or one of the devices involved are marginal in nature.

7. Out of sequence packets are not, usually allowed, meaning a valid packet may have to be retransmitted because an earlier packet got lost.

Datagram Type Net-works

A datagram type network operates in a different manner than a connection type network. Each packet in a datagram network contains a header that should have all the information necessary to get it from its source to its destination totally independently of all packet:; sent; before or after it. A datagram network is like sending a bunch of letters to the same person on different days from the same post office-; Just because the same post office was used, and the addressee is the same, doesn't mean all letters follow the same path from you to the destination.

The basic advantages and disadvantages of a datagram network are just the opposite as those of the connection network. While every packet can be routed a different way (potentially going around trouble spots in the network), the added size of each packet (due to a larger header) and the added complexity of the transport layer (to re-align out of sequence packets) add up to more overall complexity in the software or hardware used to implement a datagram network.

History

When AMRAD first started looking into the layers higher than layer 2, we were sold on the datagram type of network. It seemed to us that the amateur radio enviroment that a network must operate in can become very unreliable (not only because the rf medium may vary dramatically, but also because of the voluntary nature of the amateurs participating). Datagrams can find their way from on end of a network to the other no matter how convoluted the network may become due to equipment or operator failure, as long as there is at least one good path to the destination.

Our initial decision to use a datagram network was quickly tempered however, when we found out how large a program would be required to handle datagrams properly. We couldn't find anyone who was operation a network level datagram service without having implemented in a higher-level language on a large mainframe computer (or at the very least a mini). Obviously, we weren't going to write a program of this size in the not too distant future.

When AMRAD got together with the New Jersey packet contigent to discuss the level 2 AX.25 protocol, we met at Telenet with Eric Scace, K3NA. Eric has worked with X.25 for quite a while (he was involved in the writing of the X.25 recommendation), and he was able to bring to our meetings invaluable insight into the inner workings of X.25. I found out there is a hugh difference between reading a protocol specification, and talking to someone about that protocol: actual implementation. Eric was able to convince us that a connection oriented network such as X.25 could be implemented properly in an amateur radio environment. He also halped us decide how to implement the X.25 level 3 protocol properly, and how to add information necessary to make the protocol operate in an Amateur Radio enviroment while still maintaining the integrety of the X.25 protocol specification. Fortunately, Eric was able meet with the packet crowd at the AMSAT sponsored get-together last October. I think that a lot of questions were answered at that meeting, and al-

though the rest of the crowd isn't convinced that X .25 is the proper way to go at layer 3, we at AMRAD have decided to go with X.25 for our local network. Gordon Beattie Jr., N2DSY, from The Radio Amateur Telecommunications Society (the New Jersey group mentioned above) is in the process of writing up the layer 3 AX.25 protocol specification now, and it will be available soon.

Basically, AX.25 level 3 follow the CCITT X.25 level 3 protocol having to do with virtual circuits. All references to datagram and permanent virtual circuits are to be ignored.

We have added two amateur radio network facilities. These facilities allow for either **explicit** route selection or **implicit** route selection. Explicit route selection is where the requesting station describes exactly the route the connection should take. Implicit route selection is where the requesting station describes where the station is, and the actual path is determined by the network.

Unfortunatly, the level 3 protocol is just too complex to present in a paper of this type, so if you are interested in AX.25 layer 3 details, I suggest you get in touch with AMRAD or Gordon Beattie, and we will keep you advised as the protocol develops.

A NEW PACKET-RADIO CONTROLLER BOARD

Terry Fox, WB4JFI
Vice-President, AMRAD
1819 Anderson Road
Falls Church, VA 22043

Abstract

This paper describes the AMRAD packet assembler/disassembler (PAD) to be released soon. It is Zilog Z80A based, uses a Zilog 8530 serial communications controller and is packaged on an S-100 pc board.

Introduction

One of the main problems with packet radio is that until recently there hasn't been a lot of hardware to support the various protocols being used. Except for a few pockets of activity, most of North America has adopted the idea of using a separate board (usually a single-board computer) to handle the actual generation and reception of frames. The first production board available to the amateur to do this was the Vancouver Amateur Digital Communication Group's Terminal Node Controller (TNC). This board was (and still is) available primarily as a bare board which had to be built up by the packet enthusiast. The board uses an Intel 8085 processor, 4k each of EPROM and static RAM, a serial or parallel device to communicate to your terminal/computer, and an Intel 8273 HDLC controller. The VADCG TNC moved the packet-radio software from the host computer to a separate board, and at the same time allowed many people to use a simple terminal with packet radio.

The next board that came out was designed by and is available from the Tuscon Amateur Packet Radio Corporation (TAPR). Its basic design philosophy is the same as the VADCG TNC in that it also handles all of the frame-level generation and reception of packets, requiring only a terminal or serial/parallel interface to a computer. Its actual hardware design is quite a bit different from the VADCG TNC, however. In addition to a different CPU (a Motorola 6809), it boasts quite a bit more memory (six byte-wide RAM/EPROM sockets normally fitted with 24k of EPROM and 6k static RAM), a different HDLC controller chip (Western Digital 1933), timed interrupts, a non-volatile memory, and a complete Bell 262 compatible modem (using the Exar modem chips). The TAPR group had time to study the VADCG TNC and made a lot of improvements when they designed their board. Another advantage to the TAPR TNC is that it is purchased as an assembled board, reducing greatly the chances of failure for the user. One of the many problems with the TAPR TNC actually has nothing to do with the board itself. TAPR has had a lot of problems getting the boards designed and into production. As of when this paper is being written, TAPR is getting the boards out to the last of their customers. This is one of the disadvantages of being out at the infamous leading edge. This also shows that the TAPR group doesn't want to send out TNCs that aren't as good as they can be, and the boards are definitly worth the wait.

The Amateur Radio Research and Development (AMRAD) group has been watching the progress of these boards with interest for quite a while now, and we figured it was time we got into the act. Most of us in AMRAD that are into the development stages of packet radio use S-100 based systems, usually with Z80 processors. So the TAPR TNC is rather difficult for us to write software for. Also, while having the modem on the TNC is nice for two-meter operation. However, when testing new ideas out for hf operation or otherwise when a different modem is needed, all the on-board modem does is take up board space.

We had some different problems with the VADCG TNC. The primary one is that the memory (both RAM and EPROM) are too small. I have modified the EPROM space to allow the use of either 2716's (up to 8k EPROM) or 2732's (up to 16k EPROM), but not many people want to take an X-ACTO (tm) knife to their board. Bill Danielson, N6FQR has modified the VADCG TNC to run 2k static RAM chips and also be able to download software from another computer to the TNC during software debugging, but here again, the board must be butchered.

We have come up with an S-100 board that contains an Intel 8273 protocol chip and some support logic so that we can write and debug software for the VADCG TNC in our S-100 systems before blowing any EPROMs for the VADCG TNC. This system works very nicely when coming up with changes in existing software, or working on some radical new idea without having to reburn EPROMs for every failure (no, I haven't written error-free code in a long time).

An additional problem with the VADCG TNC is that the baud rate on the packet channel is hardware selected and it goes down only to 600 bauds. Another modification was made to allow the baud rate to be software selectable and to allow the slower speeds needed for hf operation.

The AMRAD PAD

The further along we went with packet radio, the more kludges we needed to make on the VADCG TNC to allow us to do the experimentation we wanted. Since the TAPR TNC was in the initial design stages and using a processor we weren't accostomed to, it became apparent to us that it would be easier to design and build a whole new TNC board.

After making the decision to design our own board, we next had to decide what to put on the board, and what physical size it should be. I'm sure that it will come as a surprise to almost no one that the board will fit into an S-100 frame, and steal its power off the S-100 bus. This does not preclude the possibility of using the board stand alone with a single S-100 edge connector to supply power if the user isn't using an S-100 system.

Basic PAD Layout

Fig. 1 shows the basic layout of the AMRAD PAD (PAD stands for Packet Assembler/Disassembler). The PAD was designed to be very flexible. In addition to allowing the user to connect to it in either serial or parallel mode (the other boards do this also) it allows for fully adjustable baud rate on both serial ports, and if necessary, both serial ports can be programmed to be HDLC channels. It also has room for controlling the speed of a multi-speed programmable modem. The large amount of memory allows for downloading and debugging of programs in the PAD rather than needing another simulator board in a larger microcomputer. The large RAM space also allows a lot more space for buffers and other storage, meaning that the PAD should be able to run more than one connection per HDLC channel (something that could come in handy in the near future). The EPROM area can programmed to accept 2716, 2732, or 2764 devices, allowing plenty of room for expansion. A detailed description of the PAD board follows.

PAD Power Supply Circuitry

The power necessary to operate the PAD board is supplied thru the S-100 bus connector on the bottom of the board. The PAD uses three voltages, +8V at about an amp, +18V at about 50 milliamps, and -18V at approximately 100 milliamps. These

voltages are regulated on the board to supply the +5 volts and + 12 volts that the rest of the board requires. The five-volt bus has two 7805, TO-220 type voltage regulators, one on each side of the board. The + twelve-volt regulators use the smaller, TO-92 -style packages. These voltages are used primarily for the RS-232 driver chips and the real-time clock.

In addition to the power supply mentioned above, there is also a battery supply on board to run the real-time clock, the standby RAM chip, and some of these devices support logic. The PAD also has a sense circuit on board to tell the clock chip when-to go into the standby mode because the power is being shut down. This is accomplished by using a micro-power op amp that monitors both the battery voltage and the +5-volt bus, and sends an active low signal out when the main power is below the battery.

Z80 CPU and Support Circuitry

The PAD board is designed around the Ziloa Z80 processor. The master oscillator is an Intel 8080 support IC, the 8224, running at 18.432 MHz. This frequency was chosen because it is a multiple of the common baud-rate frequency of 1.8432 MHz, and also because it is high enough to be used in the refresh logic for the dynamic memory. The master oscillator is divided by five to produce a 3.6864-MHz clock for the Z80 CPU, along with the rest of the board. 'This frequency is a multiple of the 1.8432-MHz clock desired, so it can be used as the clock input to the serial-interface chip.

The Z80 reset logic consists of one half of a dual D flip-flop to make sure that all devices using reset get a properly timed signal. The Z80's NMI pin is optionally connected to a pushbutton to allow a way of interrupting CPU operations for debugging.

The Z80's RD and WR signals are buffered, since they are fed to many of the other devices on board. IORQ and Ml* are ORed together to produce INTA* that is used by the 8530 interrupt support logic.

EPROM Memory and Support Logic

There are four 28-pin sockets provided for EPROMs on the PAD board. They have been placed so that Textool Zero-Insertion-Force(ZIF) sockets can be used if the EPROMs are going to be changed frequently (they are recommended). A jumper area is provided to allow the EPROM space to be programmed for 2716, 2732, or 2764 type EPROMs. If 2716's are used, 6116 type RAM chips also can be plugged into the EPROM sockets, allowing up to 62k of RAM and one loader EPROM to be used. This mode of operation is advantageous when testing software written on a larger system and downloaded to the PAD. It also comes in handy when debugging the dynamic RAM since the static RAM in the EPROM socket will allow monitors or memory testers to run properly.

A 74LS138 decoder is used to provide the chip enable to the EPROMs. There is also a signal generated called PROM*, which is used to automatically deselect the EPROM memory space from the dynamic memory.

Dynamic Memory and Support

The dynamic memory consists of eight 4164 type chips for a potential of 64k of memory. When this board was in its initial design stage, 4116 type devices were used since they provided a lot of memory for a very little amount of money and board space. As the PAD design progressed; the prices of the 64k chips dropped down to where it became cost effective to put them on instead. The refresh support logic is the same for both types of devices, and the power supply considerations are simpler for the 64k chips. The 64k devices also generate less noise on the power bus.

The sixteen address lines from the CPU are multiplexed onto the eight address lines of the 4164's with 74LS157 multiplexers driven by a select timing signal. This select signal,RAS*, and CAS* signals are generated by three D-type flip flops that are driven by the two clocks mentioned earlier, Ml*, MREQ*, and RFSH* signals from the Z80 CPU, and some additional logic. The PROM* signal mentioned earlier is added into the refresh logic to deselect the dynamic memory when an EPROM might be addressed instead of RAM.

The refresh logic is of standard design taken right out of the Zilog dynamic memory interface application note. The only differences are the addition of the two added address lines into the address multiplexers, and the use of a times five clock instead of a times two clock. This higher clock rate is actually better because it causes the refresh signals to be within tolerences where a times two signal would cause some timing signals to be slightly out of specification.

Input/Output Decode Logic

Most of the port decode logic consists of one IC, a 74LS138 decoder. It sends chip-select signals to the various I/O devices, except for the real time clock and standby RAM chips. Because the real-time clock and stand-by RAM chips use many more ports than the other devices, they have separate chip-enable logic. The following is a brief chart of the port assignments on the PAD:

HEX ADDRESS	DEVICE
cm-03	8530 SCC
04-07	Terminal PIO
08-0B	Auxilliary PIO
0C-0F	Clock Interrupt Latch
10-13	Standby RAM Latch
40-5F	Real Time Clock Chip
80-FF	Standby RAM Chip

8530 and Support Logic

The two serial channels are generated by a relatively new device to the amateur packet enthusiast, the Zilog 8530 Serial Communications Controller, or SCC. When the PAD was in its initial design, it used the Intel 8273 protocol chip. This was the chip that most of us were familiar with, and software already existed to drive the 8273 properly. When we discovered the 8530, we realized that it could greatly simplify the PAD. Not only does it have two completely programmable channels, it also has two separatly programmable baud-rate generators, and two DPLLs. This one chip can support both the HDLC packet channel and the serial channel to the terminal or computer. In addition, if the terminal or computer is interfaced to the PAD board thru the parallel port, both serial channels can be used as HDLC packet channels. This might be useful when the PAD board is used with a larger computer as a gateway between two net-works. or if one channel is assigned permanently to one type of operation (say hf), and the other is used for a radically operation (say vhf or phone line interface to Telenet or ARPA).

Even though the SCC is designed by the same company as the Z80 and it is supposed to be compatible, there are some timing problems between the SCC and the CPU. The main problem has to do with interrupt-acknowldgement timing. The SCC is designed to work on the Zilog Z-BUS, and its daisy-chain interrupt structure is slightly different than the Z80 peripherals used on the rest of the PAD board. This timing problem is cured by adding a couple chips that are used to extend the time of an interrupt-acknowldegenent cycle, and also allow the SCC to be properly reset" by -hardware. The timing of this delay is accomplished by waiting the Z80 CPU with a delayed signal from a 74LS164 shift register. As with the dynamic refresh logic mentioned above, this modification is right out of an application note from Zilog.

Since both of the serial channels can be used as HDLC channels, it was decided to put timers on the RTS signal of both channels. These are 555 type timer designed to time out in about 30 seconds and stay off until reset by RTS changing back to an inactive state. The timer output of each of these time-out circuits are fed back into the CPU thru two bits on the auxilliary PIO, allowing the PIO to be programmed to generate an interrupt anytime the timer changes state. Not only does this give a fairly positive indication that the transmit command was successfully accomplished, but it also gives the CPU an indtcation if a time out has occured, and allows for a fairly graceful recovery (as opposed to hitting the reset button) after a time-out situation. These timers can be jumpered around in both channels if they aren't necessary.

The auxilliary PIO output side generates two

signals that can be jumpered into the transmit data signal from the SCC. These signals are useful for adding a cw i-d by changing the connected modem between mark and space independantly of the SCC. This means a cw i-d can be generated without having to change the SCC's operating mode and kludging the software as is done on the VADCG TNC. Hopefully this will simplify both the SCC support software and the cw i-d routine.

The rest of the SCC support logic consists of TTL/RS-232-C interface chips to allow the SCC to communicate with modems or other RS-232-C type devices.

The connectors used for the serial channels are the standard 26-pin insulation displacement connector (IDC) wired so that when crimped to a cable with a DB-25 on the other end, the RS-232-C signals will come out on the correct pins.

PIO Circuitry

The PAD has two Z80 PIO chips on it. One is used for an optional terminal or computer interface if the user wishes to communicate with the PAD in parallel rather than serial mode. It can be left off if parallel interface is not required. A jumper is provided to continue the interrupt daisy-chain if this chip is not installed. If it is being used, its output signals are fed thru a 74LS244 octal buffer to provide enough drive to run the signals through a relatively long cable. The connector used for the parallel interfaces are the standard 26-pin IDC connectors and are wired the same as most parallel ports using this connector.

The auxilliary PIO is used to generate and receive several signals used by the PAD board. The parts of the aux PIO not needed for on-board functions are fed to a standard 26-pin connector, so they are available to the user for whatever he wants. In addition to the lines mentioned above that are used by the SCC interface, three lines from the output are sent to the channel A HDLC interface so they optionally can be used to control the speed of a multi-speed modem, such as might be used on hf.

Real-Time Clock and Support

The real-time-clock chip used is the National Semiconductor MM58167. In addition to keeping the time of day, it can generate timed interrupts to the CPU. and it also has an alarm function, allowing it to wake up the rest of a system at a preset time. Power for the real-time-clock chip is obtained from the plus twelve-volt bus thru a zener diode regulated power supply, some switching diodes and a battery supply. This allows for more time to cleanly shut down the system and prevent the clock chip from losing its memory.

The clock chip is not part of the Zilog Z80 family, and it does not generate any interrupt vector when it interrupts the CPU, so an octal tri-state latch was added to the board to provide an interrupt vector when the clock generates an interrupt. This vector is loaded by the CPU, so it can be programmed to be just about anything that the CPU will recognize.

The clock chip's power-down interrupt* is an open-drain output that is fed to the aux parallel 26-pin connector, allowing this signal to be used by external logic to power up a system.

Standby RAM Logic

The standby RAM is a 6514 low-power CMOS static RAM (organized as a 1024-by-4-bit device) that is addressed as a series of input/output ports. The upper three address bits to the RAM chip come from a four-bit latch. This allows the use of all 1024 nibbles in the RAM chip. The RAM is automatically deselected on power down, preventing it from being glitched by power transitions. The RAM can be used to store the amateur call, speed on both serial channels, whether the terminal/computer is using a serial or parallel channel, and many other PAD attributes thst should be saved during power down. Using a battery-backed-up RAM instead of a NOVRAM allows the RAM to be updated much easier and more often, in addition to simplifying the hardware design.

Software Considerations for the PAD

The first software running on the PAD is a Z80 monitor modified to run on the PAD. The software required to run the PAD on AX.25 level 2 is being developed right now, and should be available just about the same time as the hardware. This software is written to take advantage of the PAD's features including the potential for downloading the protocol program and also allowing multiple connections. This software is taking a little longer to develop primarily because it is being written for a TDL Z80 assembler rather than using a higher-level language.

The Zilog 8530 SCC has sixteen write and nine read registers internally for each of the two serial channels. They are addressed by first writing a pointer to the control register of the channel of interest, then the requested register can be accessed. These registers allow each serial channels attributes to be programmed, from whether the channel is to be synchronous or asynchronous to the type of flag character used. The registers have to be programmed in a certain order to obtain proper operation, especially when using the SCC in HDLC mode. For more information on the SCC and its internals, see the AMRAD Newsletter for January of 1983.

The National MM58167A clock chip has 24 internal registers available to the programmer. Eight of these registers are the counters that contain the time, another eight contain the "alarm" time, and the rest are used to control or read the various internal functions of the chip, including such things as the interrupt handling, individual counter resets, and the standby interrupt (power down alarm) control.

The parallel input/output chips are standard Z80A PIO's. They have four modes of operation for each of the two parallel channels per chip (except for port B not being able to run mode 2. The four modes are:

- Mode 0. Output mode.
- Mode 1. Input mode.
- Mode 2. Bidirectional mode.
- Mode 3. Control mode.

The terminal PIO is capable of operating in any of the four modes with no problems. The auxilliary PIO is used to control and monitor several of the internal PAD operations, along with having a few bits left over for such things as modem-speed control. The aux PIO has the outputs of the transmit timers going to it for example, so if port A of the aux. PIO was put in mode 3, it could monitor the timers and generate an interrupt whenever the timers output changes state. The PIOs do generate Z80 style interrupt vectors, so this could be accomplished very easily, allowing an elegant recovery from a time-out condition.

PAD Availability

The PAD will be available only as a bare board for a while, due to capital limitations. I will be able to offer some of the harder-to-find components on a sporadic basis for a while. Hopefully we will be able to support the board better in the near future. Anyone interested in this board should write to me at the address listed above.

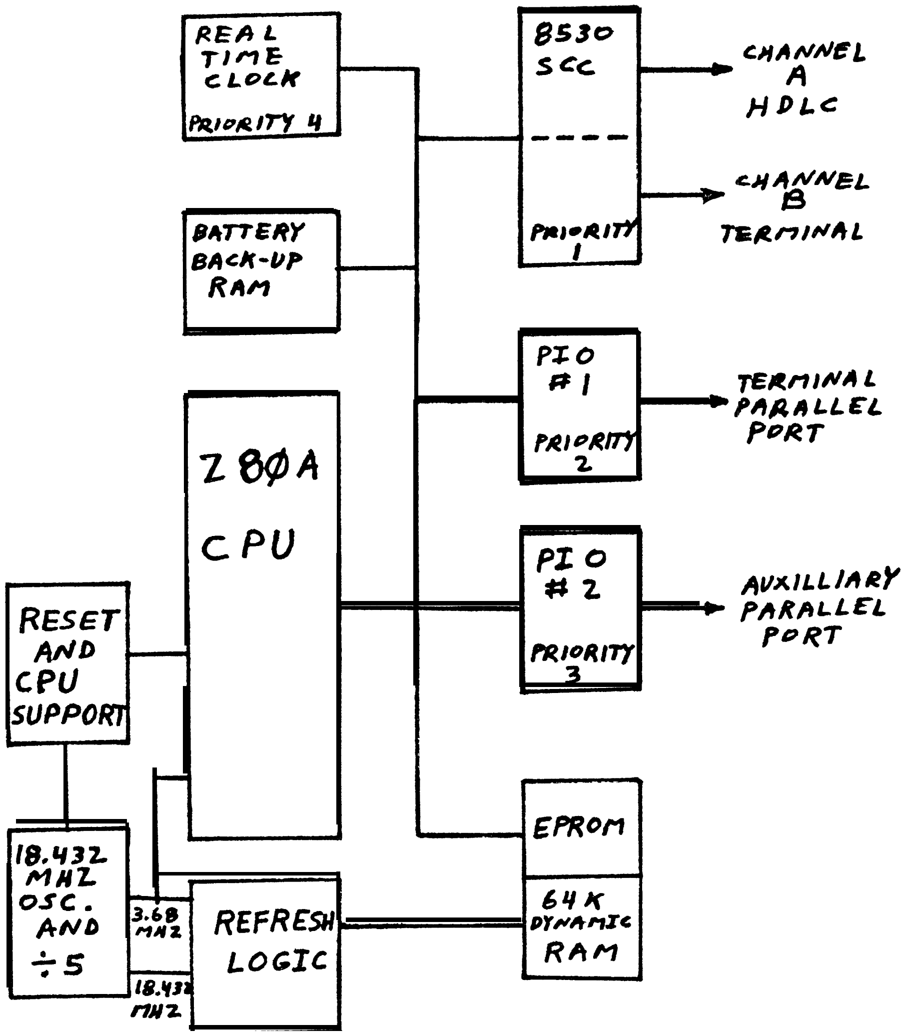


Fig. 1 - Block diagram of PAD.

DESIGN DECISIONS FOR THE TAPR TNC LINK LEVEL

David Henderson, KD4NL
2621 W. 164th St.
Torrance, CA 90504

Abstract

The decisions that were made up front on the software side of the TAPR project have had a very strong impact on the implementation and success of that project. Following is a review of some the design decisions that were made long before coding started, and a chronicle their impact upon implementation and performance.

Introduction

Before the software description starts, lets run through a quick overview of the hardware. The microprocessor is a Motorola 6809; the memory complement is 24 kilobytes of EPROM, 6 kilobytes of RAM, and 32 bytes of electrically erasable ROM. The peripheral chips consist of a Mostek 6551 asynchronous interface adapter, and a Western Digital 1933 HDLC interface chip. There is also an Mostek 6522 for clock support and onboard parallel I/O. The potential of using a parallel port for terminal I/O is provided with a Motorola 6820 parallel port chip. The computer system used as a software factory for the onboard software was an HP-64000 microcomputer development system present at the University of Arizona. This system made possible the installation of the TNC software on the TNC hardware.

The entire software development cycle was focused on X.25. The AT&T BX.25 document was taken as a reference for LAPB (reference 1), and the AMSAT document (reference 2) was taken as a reference for the header construction. The transition tables in the BX.25 document were followed very closely for the connect/disconnect sequences, but the I-frame manipulation was implemented in other ways, as described in more detail later on.

Architecture

The number one design choice was to write as much of the TAPR code in Pascal as possible. Pascal was chosen because it is a widely available high level language, and the existence of sophisticated compile time options for debugging implied the Pascal checkout could be started either on a big timesharing system or a microcomputer Pascal system.

The choice which influenced the rest of the implementation the most was to have the high level Pascal code sit in a tight loop checking flags that are continually being set and reset by interrupt code written in assembly language. This design decision paved the way for implementation and checkout of the Pascal source without having to have the hardware actually running, and greatly simplified the design of the Pascal code since it did not have to deal with interrupts. There was also a complicating factor in that flags are continually being set and reset to schedule future actions in the Pascal code; that is, the main loop of code was a sequence of IF and CASE statements that check these important flags. The action taken by the code is not immediately apparent upon reading the code; you have to know the meaning of the flags being manipulated,.

The next design hurdle was the buffering; how many buffers should there be and what mechanisms were to be used to move data from one buffer to another? The main task, being a TNC, needed only two buffers, one for the incoming data flow and one for the outgoing data flow, and it was thought at one time that only two buffers would be needed. Once digipeating and control functions were considered, it was clear that four buffers were needed - there had to be a way to shuttle HDLC input data to the HDLC output queue and a way of talking terminal input data for commands, acting upon the commands, then printing a response to the terminal. The software was then broken up into four distinct sections; each section moves data from one buffer to another, with the possible side effects such as parameter changes. One section moved data from the HDLC input ring to the terminal output ring, another moves data from the terminal input ring to the HDLC output ring, a third took data from the terminal input ring and produced status messages in the terminal output ring, and the digipeating process move HDLC frames from the HDLC input ring to the HDLC output ring.

The next choice was to have the software 'know' as little as possible about the half duplex nature of the radio link. Previous implementations such as Vancouver Area Digital Communications Group protocol (VADCG for short) had used the poll/final bit in messages to turn the link around and have the receiving side turn into the transmitting side. I did not fully comprehend the use of the poll/final bit in

this manner, and this use of the poll/final bit certainly conflicted with the LAPB usage. The design decision chosen for this "when do I send" problem can be simply stated: Only acknowledge receipt of messages or send messages when the modem signal data carrier detect is not present, or the data carrier detect signal has dropped at least once since receipt of a message. This rule means that message acknowledgements will be generated once for every sequence of information frames, and that there is a break in the received traffic before any new messages are queued up for transmission.

Consequences of the architecture

The Pascal implementation on the HP-64000 development system was not a full ISO standard Pascal. The major difference dealt with character strings; the HP Pascal had a STRING data type whereas the IS0 standard has only arrays of characters. Unfortunately when the HP system writers adopted the STRING data type, they threw out completely any compatibility with ISO standard programs - character strings longer than one byte could not be used. This problem was 'solved' by including all character strings into one area of memory in EPROM. All references to constant strings had to reference the name of the array containing the data in this read only data area.

The Pascal code was checked out on two different computer systems prior to being installed on the TAPR board. The systems were a 36 bit mainframe and an 8 bit micro. The 36 bit mainframe checkout used routines to loop back HDLC input and output together in software; this allowed the software to connect to itself and allowed the basic logic to be checked out; all of this checkout was greatly speeded up via the symbolic debugger on the mainframe, The 8 bit micro allowed on the air tests with VADCG boards, and was invaluable in shaking down more basic logic problems. Again the routines to interface to terminal and HDLC I/O had to be changed to reflect the routines existing in the 8 bit micro system, but this is a pretty easy task to accomplish, The result of this staged checkout was a very robust implementation of X.25. . There were bugs in the Pascal code, but they were only evident under extreme conditions.

Implementation of the architecture

There are four streams of data flow: Two directions for HDLC I/O and two directions for terminal I/O. Knowing when there is data present in the input streams and when there is enough room in the buffers for more characters in the output streams is handled by global variables. These variables are generally set by interrupt routines and reset when low level routines called by Pascal manipulate data associated by the flags. Each data stream has its own peculiarities, and the flag checking/clearing activity associated with each peculiarity will be covered.

The first stream of data is the HDLC input stream. Here a global variable exists which always contains the HDLC input top frame size, and is zero if there are no HDLC input frames placed into the HDLC input buffer and not yet processed by the Pascal program. The portion of the Pascal code that handles HDLC input notices that the variable is nonzero, and calls a low level routine which will move the HDLC data from the input ring buffer into a private Pascal buffer for further examination.

For HDLC output, there is a global variable which contains the number of free bytes in the HDLC output ring buffer, This cell is checked before any HDLC output frame is generated, and if there is not enough room in the HDLC output ring buffer for the potential output frame, then the generation of the output frame is deferred by simply not clearing the flags that cause the output frame to be generated. In the case of digipeated frames, if there is not enough room in the HDLC output ring buffer for the digipeated frame, then the digipeated frame is simply forgotten.

Terminal output is similar to HDLC output, but there are differences. There is a global variable which contains the number of bytes currently available in the output ring buffer. The routine called to queue up terminal data will always wait for space available in the output ring 'buffer to queue the character, The purpose in having the variable output ring free byte count is to avoid waiting for buffer space when X-frames come in on the HDLC input port. When an I-frame comes in that has a data portion too big to buffer, the data in the frame are ignored and the HDLC message "RNR" is scheduled for future transmission. This async output routine is freely callable from anywhere within the Pascal code, and does get called as a part of parameter displays, hex dumps., and internal debug routines, It was decided that when any of these activities were going on, no one would care if the Pascal code was spinning while waiting for more room in the async output buffer.

NOW we come to the most interesting of the buffering problems, terminal input. The nature of X.25 is that there can be up tO seven complete I-frames in flight at once. From the tight RAM limitations, it was clear that the input data for these I-frames could not be duplicated anyplace else in memory. The solution chosen this time around was to build a table describing the active portion of the terminal input ring buffer. The table is eight entries long, and each entry consists of a data start pointer into the terminal input ring buffer, and a data length from that start point. The table is eight entries long because that is the modulus for the X.25

sequence numbers, and these X.25 sequence numbers serve as indexes into the table. Part of the routine activity is to check to see if a new I-frame can be generated, and if so then a check is made for data to fill the I-frame. This checking is performed by calling a routine which returns a removal pointer and a size for data (if any) present within the terminal input ring. If there are data present, then the size will be nonzero, and another table entry can be constructed to describe an P-frame in flight. When I-frames get acknowledged the data space occupied in the terminal input ring buffer is marked no longer in use by advancing to the next sequence number and by changing a pointer, allowing reuse of the memory. One consequence of this design choice on input buffering is a "selective reject" (asking for fills), becomes a more difficult job to implement than if the other feasible approach of using linked lists had been made (It should be noted that selective reject is not part of X.25).

The next area that was simplified by the design decisions is the generation of HDLC frames for information transfer. There is a definite priority in X.25 for the kinds of frames that have to be sent, The priority scheme is implemented by the order in which flags are checked. These flags are generally set in the HDLC input routine, but may be set by anyone to schedule HDLC output. The flags that are checked and the order in which they are checked are: The send RNR flag (set when terminal buffer space gets low) to send a RNR frame, the send REJ flag (set when an out of order frame is received) to send a REJ frame, the received RNR flag must be reset (set when an RNR frame is received) to send an I-frame, the send RR flag (set when I-frame received) to send an RR frame. This section of code is also where the half-duplex decision must be made. The code to generate these output frames is only executed if the modem signal DCD (data carrier detect) is low or if the DCD signal dropped after any of the flags scheduling RNR, REJ or RR frames were set. This simple test is all it takes to prevent the sending of an RR frame for every I-frame received. Notice also that with the order in which the flags are checked, sending an I-frame will be attempted before sending an RR frame, so that if both sides have I-frames to send, then there are no RR messages sent when an I-frame would also acknowledge receipt to the other side.

Another detail that was made quite easy by the "no interrupts in Pascal" decision was the handling of multiple software timers. Actually, there are only two timers, the beacon timer and X.25 timer t1, but they are handled exactly the same way. The generalized timer code is implemented via a Pascal structure; this structure contains an expiration time and a Boolean flag that indicates whether or not the timer is running. "Time" is used loosely, for the only way the Pascal code is aware of the passage of time is by looking at yet another global variable. The time global variable is incremented at a one per second rate by an assembly language interrupt routine. Whenever a timer needs to be started, its expiration time is set to the time global variable plus the number of seconds in the timer interval, and a Boolean flag is set within the timer structure to indicate the clock is active, The big loop of code that is continually checking flags now has to check the timers for expiration, and this is quite easily done by comparing the global variable time with the expiration time in the timer.

Debugging and checkout

A subset of ISO standard Pascal was selected which would *work* both with the HIP-64000 compiler and the two systems that I had available for writing and checkout, The real reason for this subset decision was to allow as much checkout of the software as possible in a friendly environment. The "no interrupt code in Pascal" decision made possible the replacement of low level routines in assembly language with routines written in Pascal which could invoke standard Pascal I/O. On the mainframe, a dummy set: of HDLC input and output routines was written to loop back the HDLC output internally (from the Pascal programs point of view) to the HDLC input. On the 8 bit micro system, existing routines for MDLC input and output were modified to use new calling sequences and the Pascal code was actually executed on the air. In both debugging testbeds, the clock was simulated by incrementing the global variable holding the second tick whenever the code in the main loop noticed the system time of day change from a previous value. These sets of routines allowed checkout of the major logic flow of the Pascal software under a symbolic debugger, This self-test arrangement was extremely valuable, and allowed about ninety percent of the bugs in the Pascal code to be eliminated under the friendly environment of the symbolic debugger. Not everything could be checked, and the things that could not be checked were not: setting flags for the low level routines (which didn't exist) or missing transitions in the state/event table that were never exercised during this self-test. One clear fallout of the debugging procedure was transportability, because the software was running on two different Pascal systems before the TAPR TNC software even met the TAPR TNC hardware.

Summary

The broad design decisions that were made before implementation of the TAPR TNC software served as an aid in inplementation, supplying a framework that would support the detailed coding process. These decisions were made in the light of previous experience (and mistakes) in the

implementation of three other systems similar to X.25. There were other choices that could have been made to supply the implementation framework, but my intent was to illustrate the decisions which made the TAPR TNC software almost write itself.

References

1. BX.25 Technical Reference@ Issue 2, June 1980. American Telephone and Telegraph Company.

2. Protocol Specification for Level 2(link level) Version 1.1, Paul Rinaldo, W4RI, et al, October 10, 1982.

Unique Features of the TAPR TNC

by Lyle Johnson, WA7GXD
c/o Tucson Amateur Packet Radio Corp
P.O. Box 22888
Tucson AZ 85734

Background

The Tucson Amateur Packet Radio (TAPR) Terminal Node Controller (TNC) began as a local project done by a handful of Tucson-area Amateurs late in 1981. The project attracted enough attention to cause the formation of a formal club as well as an enlarged number of participants. As interest continued to grow, TAPR incorporated as a non-profit R&D group, and the TNC project changed from a local effort to include active participation in the design, implementation and testing phases with Amateurs from the West Coast to the Northeast.

The original Alpha-level TNC, of which only a dozen kits were distributed, led to the development of the current Beta-level TNC, now undergoing testing in dozens of sites, with site sizes ranging from three to over twenty stations. The total Beta distribution is approximately 175 preassembled TNCs.

While there have been, and continue to be, many other methods of getting involved with Packet activity, the TAPR TNC offers a number of unique features that are worth noting. It is the intent of the author that this paper accent those features, and it is the hope of TAPR that other exnerimenters may benefit from the experience and insight gained by the TAPR effort.

System Design

The TNC was designed as part of a system to allow an individual Amateur station to operate using the Packet mode with a minimum of integration problems. To this end, the required radio and terminal/computer interfaces are as general as possible. All I/O areas of the physical PC board surround a user wire-wrap area to allow for custom configuration. Further, the wire-wrap section has all significant power supply rails bordering one side.

Since it was designed as a system component, a radio-oriented MODEM (modulator-demodulator) is included on-board. To minimize the software design effort that might otherwise be required, an on-board microcomputer was designed to handle all aspects of lower-layered protocol implementation, with special emphasis in hardware to allow operation in an Amateur radio environment,.

In any microprocessor- based device, software is a very important consideration. Consistent with the TAPR philosophy of modular flexibility, considerable efforts were made in hardware design to accomodate the desires of the software groups. Parallel software efforts were made, with the Pascal-based implementation of the "standard" protocol adopted in Washington on October 10, 1982, being initially distributed with the Beta TNCs. A FORTH- based design, implementing a dynamic addressing level-two protocol, -is in process of being integrated on the TNC for the Beta testing phase.

Without proper documentation, most equipment is either rendered useless, or requires an inordinate amount of operator interaction to discover its "secrets." TAPR determined early on to provide as complete a manual set as was practical.

Finally, since it is generally conceded that Packet radio is in its infancy, the TNC design group made particular efforts to ensure flexibility as well as capacity for expansion.

MODEM Design

The MODEM incorporated in the TNC is designed for compatibility with the de facto standard Bell 203 tone pair, 1200 and 2200 Hz, using simple AFSK at a maximum data rate of 1200 baud.

The tone generator is straightforward, using the Exar 2206 FSK IC in a low-distortion, sine-wave configuration. The output is buffered by a simple +1 op- amp circuit, then ac- coupled to the radio interface connector. The output amplitude is user settable via a twenty- turn trimpot over a range from millivolts to volts.

During early experiments with the TNC, it was discovered that the WD 1933 HDLC controller did not allow direct control of the output state of its Tx Data pin when used in the NRZI mode. This complicated the issue of the US-required CW identification. In the end, use was made of the Exar 2206's analog multiplier input to allow for tone on-off keying of the CW ID. This input is buffered by a TTL inver-

ter, allowing the tone state to be set under software control, independent of any other MODEM parameter.

The tone generator frequencies are easily changed via adjustment of individual twenty-turn trimpots. To facilitate such adjustment, the TYC hardware includes, and the present software supports, a frequency counter. This is a programmable 16-bit counter/timer used by the calibration routines as a period measurement timer. The software is designed such that the **user** may **calibrate almost** any frequency within the range of 10 Hz to over 230 kHz. The resolution is +/- 1.08 uSecs (one cycle of the system **master** clock of 921.6 kHz).

Although not **strictly a** MODEM function, control of the transmit control line is a necessary feature of the radio transmitter interface. To help prevent a "glitch" from allowing the TNC to lock up a radio frequency by continual assertion of the transmit control line, a monostable multivibrator, or "one-shot", is in series with the software- controlled transmit command line. A time constant of about 14-seconds allows a multiple- frame Packet transmission at the "normal" VHF packet rate of 1200 baud, along with a CWID of a typical Amateur call sign at 20 WPM. The particular circuit implemented utilizes a 555 timer IC in a non- retriggerable configuration. Non- **retriggerable** simply means that the 14- second timeout can't be held "on" by a continuous input -- the input must be removed before the 555 **will** accept another trigger. Thus, a Packet "channel" is protected from a runaway transmitter with minimal impact on system performance.

On the receive side, an **Exar** 2211 phase-locked loop (PLL) FSK demodulator is used. This circuit has a wide dynamic range input circuit (3 mV to 3-volts), and features both data and "carrier detect" outputs. (Note that the **carrier** referred to is the audio subcarrier, not the rf carrier.)

The circuit parameters have been optimized for operation at a data rate of 1200-baud, utilizing simple FSK with tones of 1200 Hz and 2200 **Hz.** After initial design, extensive testing was conducted and the circuit values "tweaked" for best operation. A few problems cropped up that may be of interest to others experimenting with similar MODEM designs.

The XR2211 chip is very sensitive to amplitude differences of the two input frquencies greater than about 3 db. At 6 db, the PLL will take so long to lock as to render the channel inoperative at the desired baud **rate.** Specifically, TAPR found that a typical 2-meter FM transceiver would not operate reliably above about 450 baud with the XR2211 used in an uncompensated circuit, while 1200 baud was easily realized in a compensated one. Various means of compensation were tried, and the best appeared to be an active filter. A CMOS switched **capacitor** filter was implemented, with a DIP header carrying the **resistor** network needed to configure the filter for a particular radio. The DIP header allows easy reconfiguration. This design is presented in detail in another paper in these proceedings.

In addition, a means of operator feedback for receiver audio level setting was deemed desireable, to prevent overloading of the filter while maintaining sufficient audio to prevent serious degradation of the s/n ratio. This was implemented with a simple back- to- back LED pair between the audio input buffer (a -10 amplifier) and the filter input network. Initial results from the field indicate this method to be both easy to use and effective.

Microcomputer

The TNC uses a 6809- based controller for logic implementation. This is believed to be the first, Amateur radio application of the 6809 apart from individual efforts. The architecture of this microprocessor (uP) lends itself readily to high level, block-structured programming environments, and the first cut of TNC software is in fact written in Pascal.

The memory subsystem of the TNC utilizes a six- site bank of 2% pin sockets configured in the JEDEC "two line control" bytewide standard. This means **that** RAM, ROM, EPROM and/or EEPROY may occupy any socket. In addition, the memory may occupy any integral multiple of 2k- bytes capacity. To allow for future memory devices of greater density than used in the current TNC configuration, the memory map was carefully planned and executed in a bipolar Shottky PROY decoder.

The initial TNC consists of 6k- bytes of RAM contiguous from address 0, 4k-bytes of I/O space starting from address 2000H, and 24k- bytes of EPROM contiguous from address 0A000H through 0FFFFH. 2k-byte RAMs are used, along with 8k- byte EPROMs. 8k- byte RAMs can easily be accomodated, and 16k- byte and **larger** EPROMs can be used with a simple one- pin jumper (the socket wiring is presently compatible with 2764 and smaller EPROMs -- the one- pin change will make them compatible with 2764 and larger EPROMs). One Beta **tester is** building a piggyback adapter to utilize all 64k- bytes of address space.

Another unique aspect of memory design is the incorporation of the Xicor NOVRAM. This is a 256-bit device that can be accessed as a normal RAM (access time 300 nSec) as well as a 5- volt only EEPROM. This allows the user to store such parameters as station call sign, serial

port parameters, HDLC port parameters, etc., and have the information retained after power is removed. However, the user can also change these parameters interactively. This provides a significant degree of freedom for the operator / experimenter, and is believed to be a first in Amateur radio.

Software Considerations

The Beta TNC is designed to support a variety of protocols by virtue of its large address space and third- generation UP. The initial Beta release software, discussed in much greater detail elsewhere in these proceed ings, was designed to be compatible with the existing "Vancouver" protocol as well as implementing the lower layers of the AMRAD sponsored "AX. 25" protocol adopted by the major US Packet groups in Washington, DC, on October 10, 1982.

The software package produced not only supports these modes, but it also supports operation of the TNC as a digipeater under either of the two protocols as well as a beacon (under either protocol). The TNC can in fact be used as all three of the above devices simultaneously, the only limit being that all functions must be under either one or the other of the two supported protocols, but not both.

Further, the software package supports certain powerful trace and debugging modes, which have proven to be of extreme value during system troubleshooting.

Other Features

The TNC design provides for software configuration of nearly every parameter associated with the I/O ports. For the serial (RS-232) channel, these parameters include baud rate, number of stop bits, parity options and data word length. For the HDLC port, the baud rate is under software control-, using a 16-bit programmable timer. This allows using nonstandard data rates, such as the 400 baud being considered for experiments on the upcoming Phase 3B AMICON channel.

The parallel I/O port, to be supported in an upcoming revision to the Beta test software, is a full handshaking parallel interface, with separate S-bit channels for input and output . Further- , an EPROM programming adapter is being readied to allow users to bootstrap themselves along in software development and distribution.

Documentation

The TNC Beta document is a very comprehensive manual, both of the TAPR TNC in particular and Packet radio operation in general.

Over 140 pages in length, it includes chapters on setting up, operation and calibration, as well. as in- depth discussions of the hardware, protocol and system design. It sets a standard seldom seen in manuals accompanying very expensive Amateur equipment, and far exceeds what many have come to deplore as "manuals" for the more common Amateur radio devices,

The Beta Test manual is currently in a state of rapid expansion as reports come in from Beta Test sites. The eventual manual will include expanded appendices with detailed information on interconnection with a plethora of commonly used &meter FM radios, as well as hardware and software installation data for many personal computers and terminals. The intent is to provide sufficient detail to allow a non-technical Amateur to successfully bring up a Packet radio stat ion wi 'thou t other assistance.

Beta Test

Perhaps the most unique feature of the TAPR TNC is the manner in which it is being tested. Many innovative Amateur devices have been built as a result of a club project, involving perhaps 20 or 30 units. Commercial interests often build prototype units and allow selected customers to test them, who then offer feedback for the final production release of the product in question.

When TAPR decided to go ahead with the TNC design and implement it, the mailbox was full of requests from Amateurs desiring to be included in the initial distribution. Realizing the tremendous resource represented by these eager participants, and recognizing that the 'local Tucson "core" would be hard-- pressed to fully shakedown, test and debug the TNC in a reasonable time frame, the decision was taken to allow a fairly large number of TAPR members the opportunity to contribute to the TNC design effort by providing a test bed.

The participants were informed that this was to be a test in 'both word and deed, and that all Beta testers were expected to file formal reports via Beta Test Coordinators to be located in each area. Further , to make the test sites as autonomous as possible, each local site was to determine its own coordinator. 4 network was established that now spans the USA. and reaches, via AMSAT- sponsored participants, to Asia, Africa, Europe and Oceania.

Beta sites were to include both technical and non-technical Amateurs, and were to be self-suffcient as to support, modification, repair and updating of the TNCs within the site. It was felt that this would both relieve the pressures on the Tucson group and provide a reasonable cross-section of Amateurs with regard to

technical expertise, climatic variation, rfi and frequency congestion (jammers and the like).

Further, by using this method, it was believed that a maximum number of Amateurs in varied locations would be exposed to the raucaus noises generated by the packet frames, and inquiries would lead to further interest in the mode. A local group could then "spread the packet gospel" and help ensure the further growth of the mode, even while it undergoes changes associated with its infancy.

Finally, many people who contacted us, and continue to contact us, indicated that there seemed to be no cohesive force amongst packeteers, nor central clearing point for information. TAPR has therefore attempted to fill this void, and the results to date have been most gratifying. By making Beta Test a national endeavor, participants are able to become involved in a broad- based, grassroots packet effort. By providing support for the TNC, people are reassured that, in the event they face real problems, there is a source for technical assistance, both in hardware and software.

The Beta Testing of the TAPR TYC is still in its initial phase, that of distribution. As of this writing, over 110 of the scheduled 170 TNCs have been shipped to a number of sites scattered across America. Yost sites have had little trouble getting the TNCs on-the- air. Many radios, terminals and personal computers have been successfully interfaced.

Already, defect reports are coming in. A few significant hardware bugs have been noted, and work has commenced on solving the problems. Software- related problems have been reported, and the second release of Beta software will be appearing around the time of this conference.

It is believed by the author that this level of testing and cooperation has not been seen in Amateur radio before. The Beta Test participants have been very enthusiastic, as well as patient and supportive of the entire effort.

Conclusion

The TAPR TNC includes a variety of unique and innovative circuits, concepts and ideas. It is designed for both the technically inclined experimenter , for whom it offers a host of features accenting flexibility and expandability, as well as for the less- technically inclined operator, for whom it offers ease of operation.

The documentation of the TNC is at a level uncommon in Amateur radio, with a depth and breadth seldom seen outside of professional circles.

The initial software release incorporates many advanced features to allow the experimenter great latitude in optimizing system parameters, as well as flexibility in his particular system configuration. Compatibility with both the "Vancouver" and the "AX.25" protocols is provided.

The method of Beta Test is unique within Amateur radio endeavors, and is designed both the enhance the TNCs suitability for Packet radio use as well as increase the exposure of the general Amateur community to the strengths and benefits of this mode.

Acknowledgments

The author wishes to express his gratitude and thanks to the more than 250 members of TAPR, without whose support this project would never have gotten off the ground; to AMRAD and the ARRL for the opportunity to present this paper; to the Los Angeles and St. Louis contingents of TAPR for help in the areas of software, board layout and transformer procurement ; and a special thanks from TAPR to Modular Mining Systems of Tucson for the extended use of their lab and other facilities, without which the TNC might never have been initially designed, nor built for so low a cost.

Modulation and Access Techniques for PACSAT

Phil R. Karn, KA9Q/2

Assistant Vice President for Engineering, Systems Analysis

Radio Amateur Satellite Corporation

ABSTRACT

This paper describes work underway within AMSAT to define modulation, channel access methods, and related system-level considerations in the design of the store-and-forward packet radio satellite known as PACSAT.

This is not intended as a comprehensive design specification, primarily because one doesn't yet exist! In particular, only those decisions primarily concerning spacecraft hardware design are emphasized here, since the details of control algorithms, protocols, etc, will reside in software capable of being changed and reloaded into the onboard computer(s) after launch.

1. Orbital Considerations

PACSAT is intended to operate in a low altitude, polar orbit with the special characteristic that the satellite is accessible from any point on earth at least twice every day, at about the same local time each day. These so-called "sun synchronous" orbits are frequently used by weather and earth resources missions. Oscars 6, 7, 8 and 9 are all in sun synchronous orbits, so many amateurs are already familiar with them.

The low altitude and relatively high velocity of such a satellite has several implications for communications:

1. For most earth locations, a sequence of two or three passes occurs twice daily, as the turning earth carries the station through the orbital plane every twelve hours.
2. Coverage at any given time is relatively small and continuously changing. A geostationary satellite "sees" a fixed portion (41%) of the earth's surface. By comparison, Oscar-8 covers all points on earth at least twice per day, but when it is directly over the north central US, it can only see North America.
3. Passes are short. A typical pass may last for only 15 minutes from horizon to horizon. For digital communications, the highest possible bit rate is desirable to maximize the amount of communication that can be carried during a pass, although for operational flexibility the spacecraft downlink transmission rate will be under the control of a command station.
4. Path losses are modest, approximately 25-30 dB lower than to geostationary satellites. This may allow the use of lower transmitter power, omnidirectional antennas, or less efficient modulation techniques, all of which help reduce costs.
5. Doppler shift is significant. At 70 cm, horizon-to-horizon Doppler shift can be as much as plus or minus 10 khz, requiring some form of automatic frequency tracking for optimum performance. Extra wide receiver filters and noncoherent demodulation will tolerate Doppler shift, but at the cost of reduced performance.
6. Propagation time is short, ranging from 3 milliseconds when the satellite is overhead to 12 milliseconds on the horizon. Therefore, a communication protocol requiring close interaction between the satellite and its users would not unduly penalize performance except for very small packets.

2. Downlink Margins

In designing a communications spacecraft, there are always practical limitations on power, size and weight, often with the emphasis on power. Hence, the power available for the spacecraft transmitter becomes the limiting factor in the overall system design. For this reason, it is helpful to start our analysis with a stated, realistic value. This immediately provides insight into the range of modulation and bit rate options

available, therefore, AMSAT's best estimate of RF output power for PACSAT is P-2 watts.

For a spacecraft in an Oscar-8 orbit (altitude 900 km), path loss on two meters would vary from 135 dB when the satellite is directly overhead to 147 dB with the satellite on the horizon, a range of 12 dB. Since even during an overhead pass the satellite spends most of its time "nearer" the horizon than directly overhead, we will be conservative and use the horizon figure in subsequent calculations.

Given that approximately 1 watt of transmitter output power is available on 2 meters, and that omnidirectional antennas are used both on the spacecraft and on the ground, receiver-input power would be -147 dBW (-117 dBm, or .3 microvolts into 50 ohms.) A receiver with a bandwidth of 15 khz and a noise temperature of 300 K would generate an equivalent. input noise power of -162 dBW, for a carrier-to-noise ratio of 15 dB, adequate for a low bit error rate with virtually any digital modulation scheme. However, since the satellite will move rapidly, use a real antenna with unavoidable pattern nulls, and often pass behind such real-world obstructions as trees and hills, additional margin is desirable.

This could be achieved in several ways, by

1. Reducing the receiver bandwidth. Each factor of two reduction would improve the carrier-to-noise ratio figure by 3 dB. However, for a given type of modulation, this reduces the bit rate, which is undesirable because it limits the amount of traffic that can be sent during the relatively brief passes.

2. Using more sensitive receivers. It is now quite easy to find inexpensive preamplifiers with low noise figures. However, external noise then becomes a factor, limiting the degree of improvement possible.

3. Using gain antennas with automatic tracking. Techniques for this have been experimented with by AMSAT members for several years, and soon will be within the realm of the average user of the Phase-3B spacecraft.

   Although the computation of antenna pointing angles has become very easy (the AMSAT ZX 81 project uses that very inexpensive personal computer for the task), the gain antennas are still relatively large and require fixed locations. We should therefore require gain antennas only as a last resort, in the sense that it should be possible to make effective use of PACSAT without them.

4. Using more efficient modulation methods. There exist techniques which can give much better performance in the presence of noise which are not yet widely used in the Amateur service. Many of these techniques have been widely used in commercial terrestrial and satellite services, but until now, they have often been considered out of reach of the average amateur because of cost and complexity, However, advances in digital electronics has de-coupled the issue of cost from complexity, bringing these techniques within reach of the average amateur.

3. Modulation Alternatives

We feel that the modulation method chosen is a crucial element of the PACSAT design, and I will spend the next part of this paper evaluating our alternatives.

3.1 AFSK-FM

Audio frequency frequency--shift-keying on an FM carrier is the technique currently in widespread use for terrestrial amateur packet radio, with the Bell 202 modem frequencies (1200 and 2200 hz) a de-facto standard. In the amateur- satellite service, UOSAT-Oscar-9 uses bit-coherent AFSK-FM for its VHF and UHF telemetry, with tone frequencies of 1200 and 2400 Hz - close enough to those of the Bell 202 to allow the use of that (non-coherent and therefore non-optimum) modem by the majority of stations receiving telemetry.

AFSK-FM has several major advantages: it is cheap, simple, and allows the use of general-purpose transceivers without modification. Doppler tracking Is relatively easy, since most amateur FM receivers have sufficient bandwidth to allow large frequency deviations (e.g, 1 khz) without significant degradation of bit error rate, and the modulation tone frequencies are not directly affected by Doppler shift.

Despite its simplicity, however, AFSK-FM has serious disadvantages for satellite use, which rule out its use in PACSAT:

1. Inefficient bandwidth utilization. A 15 khz channel is used by UOSAT to carry a (maximum) 1200 bps data stream, a bandwidth density of only .08 bits/hertz. Particularly

in the 2 meter band, spectrum efficiency is an important consideration.

2. Poor noise performance. Since AFSK-FM is essentially doubly-modulated FM, it exhibits a very sharp noise threshold at a relatively high carrier-to-noise ratio, and suffers greatly from impulse noise. Subjective experience with reception of the 350 milliwatt 145,825 MHz UOSAT-Oscar-9 telemetry beacon shows that pulse noise, e.g., from power lines, causes significant errors even at signal strengths otherwise sufficient to cause "full quieting" in the baseband FM channel. Local impulse noise and fades below threshold due to spacecraft rotation and polarization losses cause many errors, despite a theoretically good link margin.

3.2 FSK

Although in common use on the HF amateur bands, "straight" frequency shift keying (FSK or F1) has not yet come into widespread use on the higher frequency bands. FSK at VHF can be implemented with simple modifications to most conventional VHF-FM transceivers; a direct input to the modulator and a slicer on the discriminator output is required. The spectral efficiency of FSK can be as high as 1 bit/Hz; for our working bandwidth of 15 khz, FSK could realistically support a data rate of at least 10 kbps.

The Bell System's Advanced Mobile Phone Service (AMPS:, uses NBFM voice transceivers with 8 khz peak deviation, 30 khz IF bandwidth, and discriminator detection to carry a biphase-encoded 10 kbps FSK signaling channel. In bipkase encoding, the data stream is exclusive-ored with a clock at the bit rate, resulting in a signal with no DC component whose energy is concentrated about a frequency equal to the bit rate.

This allows an "indirect FM" (phase modulated) transmitter to be used, provided an integrator is inserted between the bi-phase encoder and the modulator. This is a direct adaptation of FM techniques used to encode data on disks and high-density magtapes. Since in FM the baseband (demodulator output) noise level increases with increasing frequency, a practical limit to the bit rate exists requiring relatively high receiver input C/N ratios when operating at high rates. In AMPS, considerable retransmission redundancy is also provided since multipath fading, not gaussian noise, is the primary source of errors in mobile radio.

AMPS is evidence that high rate FSK is practical given a sufficient C/N ratio. Performance would be better with PACSAT, since multipath fading is less severe in satellite channels than in terrestrial mobile radio, except when the satellite is near the horizon. For our working C/N figure of 15 dB, noncoherent reception of FSK (e.g., with an ordinary FM discriminator) would provide a theoretical bit error rate of about 10^-7; acceptable, but with little margin for impiementation losses and fading. For example, if C/N dropped to 12 dB, the bit error rate would jump to 5 x 10^-4.

Because of the tight C/N margin, Doppler correction is required in order to allow narrow receiver bandwidths no wider than the signal. Since biphase encoding produces no baseband DC component, Doppler could be tracked by a simple integrator connected to the discriminator output.

We can rule out use of noncoherent FSK modulators for PACSAT, because it will be shown later that another modulation technique (MSK) exists which is compatible with simple FSK demodulation but also allows coherent detection with a 3.5 - 4 dB improvement in bit error rate. However, our analysis of noncoherent FSK is useful because it indicates the performance that could be expected if MSK is demodulated with an FM receiver.

3.3 DSPK

Differential Phase-Shift Keying (DPSK) is relatively new to amateur radio, but will be used by AMSAT for the Phase III spacecraft engineering beacon. DPSK has significant advantages: it is fairly bandwidth efficient, works very well in low carrier-to-noise levels, and can automatically track Doppler shift if correctly designed.

DPSK is actually a modified form of true PSK in that the change (or- lack thereof) in carrier phase between each bit interval is used to determine the output state. In true PSK, the absolute phase of the carrier during each bit interval determines the output state, which requires an absolute phase reference at the receiver. If a clock is derived from the incoming data stream, there would be a 50-50 chance that the receiver would synchronize 180 degrees from the correct value, resulting in a 100% bit error rate! Differential PSK avoids this problem at the cost of having channel errors "propagate" through successive bits. However, in a packet environment where only a single bit error is needed to cause rejection of a packet and retransmission, extra errors "caused" by the first are of no consequence.

The noise performance of DPSK is considerably better than conventional FSK; for our 15 dB reference C/N, the bit error rate would decrease to 10^-10. However, the real advantage would be under marginal conditions: the bit error rate would not increase to 5 x 10^-4 until the C/N ratio had decreased to about 9 dB. Within a 15 kHz bandwidth DPSK could carry 15 kbps, although its noise performance would be degraded by such tight filtering; 9600 bps would be more realistic.

AMSAT has considerable experience with DPSK modulators and demodulators designed for 400 bps telemetry reception from Phase 3-B. Experiments have shown that non-linear transmitters are OK, and that 2.4 khz SSB transceivers are fine as long as compensation is made for the nonlinear phase response characteristic of SSB IF crystal filters.

The Phase 3 telemetry decoders use Costas loop carrier recovery which provides optimal performance, but may take an excessively long time to lock up in a multi-access packet environment. However, very simple DPSK demodulators exist that require no clock recovery circuit and are able to lock up in essentially a single bit time. These methods work at the expense of noise performance; the figures quoted above refer to this form of demodulation, while the Phase 3 demodulators do somewhat better.

NASA has also made extensive use of low cost, low-speed, (100-400 bps) doppler-tracking PSK systems with low altitude satellites for applications including remote data collection and search-and-rescue.

3.4 MSK

Minimum Shift Keying (MSK) is a hybrid of FSK and PSK. It can be regarded either as coherent FSK with a shift of exactly one-half of the data bit rate, or as PSK where the modulating waveform is a triangular ramp produced by integrating the binary input signal. Another equivalent way to look at MSK is as quadrature PSK (PSK with four possible phases instead of two) in which the quadrature channel carries the same data as the main channel but delayed by one-half bit period. In fact, the usual method for optimally decoding MSK involves building two PSK demodulators with combining circuitry; clearly this is more involved than a simple PSK demodulator.

One of MSK's advantages over PSK is that is requires minimal filtering to reduce its bandwidth to the minimum required. It has a constant envelope amplitude, unlike bandwidth limited PSK, allowing it to pass through a real-world linear spacecraft transponder with minimal intermodulation distortion to other signals, It can also be passed through a nonlinear (e.g., Class C) amplifier without the envelope distortion and resultant bandwidth-spreading that occurs with DPSK signals. The other advantage of MSK, perhaps the major one for our application, is that it can be decoded with simple noncoherent FM discriminators with a theoretical 3.6 dB loss of noise performance.

Optimally decoded MSK and PSK have almost the same performance in the presence of Gaussian noise. However, MSK has a significant advantage over PSK in cases of adjacent channel interference, due to MSK's smaller bandwidth. Tighter IF filters can be used with less performance degradation, and it should be easier to attain a rate of 15 kbps in our 15 khz bandwidth. Differential decoders eliminating the need for carrier recovery, similar to those mentioned earlier for DPSK, exist for MSK but are not as simple.

3.5 Discussion: MSK vs. DPSK

The "votes" are not yet all in among the PACSAT system definition and design team, although we have narrowed the choice to one between DPSK and MSK. MSK's most significant advantage is clearly its compatibility with simple noncoherent demodulators such as an FM discriminator,, However, this penalizes those who want to "do it right", as optimal demodulation of MSK essentially requires building a PSK demodulator twice.

We could go with a form of DPSK essentially similar to that used by the Engineering Beacon on the Phase 3 spacecraft, except at a higher bit rate. While there is strong interest in MSK for AMICON (Phase 3) data communications, the relatively short time available to settle major hardware-related PACSAT issues could cause us to choose the simpler technology, i.e., DPSK. While the FSK demodulator compatibility feature is no doubt attractive, optimal demodulators for PACSAT would be produced by AMSAT on a relatively large scale, and would probably be a small fraction ($50 - $100) of the total station cost.

Either method would provide means for Doppler correction. The Phase-3B telemetry receivers use Costas loop demodulators for the DPSK signal, generating as a byproduct a correction voltage indicating the offset of the downlink carrier. This correction voltage can be taken out of the receiver and applied with the appropriate amount of gain to the transmitter, tuning its frequency to compensate for uplink Doppler. While the uplink channel demodulators will probably be able to

track out the frequency shift without correction, we feel that minimizing channel lockup time is important enough that correction should be provided.

4. Access Conflict Resolution

PACSAT will be a multi-access satellite, intended to serve a number of users simultaneously attempting to send messages to the satellite. The downlink transmitter will be connected to the onboard computer, not directly to the uplink receiver as in conventional "bent-pipe" satellite transponders. Despite the short propagation delay, users will not be able to monitor the immediate status of an uplink channel by listening to the downlink, as it may be busy sending down a message intended for another user. Therefore, provisions must be made to resolve uplink access conflicts. (Naturally, since only one transmitter, the satellite, transmits on the downlink, access resolution is relevant only for the uplink.)

Assume for the moment that the satellite traffic will be "balanced", that is, the amount of traffic successfully received at the satellite will be approximately equal to the amount of traffic sent back to the ground when averaged over a sufficiently long period of time. It is agreed that this is an unlikely situation, which would only be true if PACSAT were to be used exclusively for point-to-point communications. Repeated transmission of the same information from the satellite (e.g., broadcast bulletins or spacecraft telemetry) would disproportionately increase downlink loading. However, it is my assertion that a balanced traffic assumption is a useful one, as it represents a "worst case" for system design.

All known methods which resolve contention between multiple uplink transmitters require overhead, and hence more bandwidth, than downlink transmissions for which there is only a single transmitter. We are therefore tentatively planning to use the 70 cm band, which has a 3-Megahertz Amateur Satellite Service allocation (435-438 MHz), for uplink transmissions to PACSAT and the smaller 2 meter band segment for the downlink.

In the following sections, I will describe two possible access methods for PACSAT, and compare their relative merits.

4.1 Pure Aloha With Multiple Uplink Channels

The Aloha method calls for each station to transmit at will, without concern to interference with other transmitters. (Since stations communicating with a satellite are usually far enough apart to prevent them from hearing each other, not much is gained by listening on the uplink frequency.) The well-known maximum theoretical throughput of an Aloha channel, above which delay time rises without bound, is 18%.

A very simple and attractive scheme therefore appears. If it is desired to balance uplink and downlink capacity, six uplink channels (6 x 18% = 108%) could be provided. Each one is "equal" to the others and scanned rapidly enough by the spacecraft's onboard computer to allow simultaneous reception, at least for a time, on all six. A user station would select one of the six uplink channels essentially at random whenever it has traffic for the satellite. Since the channels are all equivalent, all that matters is that the stations distribute their traffic across the channels in order to level out loading. This could be accomplished simply by allowing each station to choose an uplink frequency at random, changing it as often as desired, perhaps with each transmission. It can be shown that with a sufficient number of stations, traffic will tend to become evenly distributed over the channels.

It should be pointed out that to provide flow control, a requirement independent of the access method chosen, the spacecraft and ground computers will follow a synchronized "handshaking" protocol once a traffic transfer starts. If the ground computers are "patient" enough, that is, they allow enough time for processing and queuing delays aboard the satellite, collisions would result only when new stations initially access the satellite.

In addition to providing flow control, the go-ahead messages to each station could include a "recommended" uplink channel to use. Based on channel loading statistics kept in the spacecraft computer, the ground station would still be free to use any channel it wished, although following the recommendation would improve uplink traffic distribution.

4.2 Reservation Aloha

The "anarchy" of the Aloha system could be reduced somewhat, with an associated improvement in spectrum efficiency, at the cost of extra discipline in the ground station computers and added delay.

One of the uplink channels is designated as the "calling channel", on which stations transmit their initial requests for service to the satellite. This is in contrast to the last scheme, in which a new station may request service at any time on any channel. Requests would indicate the amount of service desired,

and because they would be short, traffic on the calling channel would hopefully be well below the 18% "total bedlam" figure. The satellite responds by granting the requesting station permission to transmit its traffic on a specific frequency during a given time "slot". Depending on the length of the time slots and the tightness of their scheduling, each station might to compute and compensate for propagation delays which change continuously during a pass.

There are two advantages of Reservation Aloha:

1. New stations requesting service would not interfere with data exchanges already in progress on the working channels.

2. Due to the tight scheduling of the working channels, fewer of them might be necessary, reducing spacecraft hardware complexity.

4.3 Discussion: Pure Aloha vs. Reservation

These two schemes represent specific points in what is actually a fairly continuous spectrum of alternatives between "total anarchy" and "total discipline". The "more disciplined" reservation scheme with the designated calling channel can potentially provide better channel utilization than the pure Aloha method; however, it suffers from an "Achilles Heel" in that it is much more susceptible to jamming, accidental or otherwise, particularly on the calling channel. With any channel usable both for calling and working, the multi-channel Aloha system provides built-in redundancy against certain hardware failures as well as jamming. For this reason, along with the strongly attractive feature of simplicity, we feel that each channel should be equivalent, although by ground software convention one channel could be used primarily for initial service requests.

While the throughput of Aloha may seem low, the 18% figure is valid only for a very large number of users; "excess capacity" exists in systems with a small number of users, especially those in which one user presents most of the traffic load. If it turns out that uplink loading becomes a limiting factor (unlikely for reasons discussed earlier), it would be possible to change operations to a "slotted Aloha" access method. This would involve programming the ground station computers to "agree" that a reference event, e.g, the beginning of a certain telemetry frame periodically interspersed into the downlink data stream, represents the beginning of a packet slot. If the ground stations were to time their transmissions to coincide with such slots, the utilization of each channel could double to 37%, and this improvement could be obtained with no changes to spacecraft hardware or software. However, each station would have to compute and correct for the varying propagation delays to the satellite as in the Reservation Aloha system.

5. Summary

This paper has presented and discussed the various modulation and access method alternatives available to the PACSAT design team. It must be emphasized that the conclusions reached here are preliminary; only after considerable simulation, experimentation, and breadboarding activity will the final decisions be made.

In any case, it is probably true that we have already "over-engineered" the PACSAT uplink in that the downlink will almost certainly become the throughput-limiting factor. Now if we only had a few more watts of power....

6. Credits

The ideas presented here actually represent those of a large number of AMSAT people in addition to the author, including but not limited to:

- Dr. Thomas A. Clark, W3IWI, AMSAT USA President and perhaps the initial "instigator" of the PACSAT concept;
- Mr. Den Connors, KD2S/7, Assistant Vice President for Engineering, Spacecraft Systems and PACSAT Project Manager;
- Dr. John L. DuBois, W1HDX, Phase 3 Ground Command Station Coordinator;
- Mr. Jan A. King, W3GEY, Vice President for Engineering;
- Dr. Karl Meinzer, DJ4ZC, President AMSAT-DL;
- Dr. Stephen E. Robinson, W2FPY, Assistant Vice President for Engineering R&D

REFERENCES

[1] *Bell System Technical Journal,* January 1979. (Special issue on the Advanced Mobile Phone Service)

[2] *Reference Data for Radio Engineers,* Howard Sams & Co, Inc.

[3] J. L. Pearce, *Measured Performance Comparison of Fast FSK [MSK] and BPSK,* Canadian Electrical

Engineering Journal, Vol. 2 No. *4, 1977.*

[4] W. B. Bruene, *MSK Simplified,* Collins Radio Company (released paper.)

[5] Mischa Schwartz, *Computer Communication Network Design and Analysis,* Prentice Hall, 1977.

[6] Dr. John L. DuBois, W1HDX. (AMSAT Phase 3 PSK notes).

[7] Bhargava, Haccoun, Matyas and Muspl, *Digital Communications by Satellite,* Wiley, 1981.

A BLOCK ORIENTED INTERFACE FOR CP/M* AND THE VADCG TERMINAL NODE CONTROLLER

Douglas Lockhart, VE7APU/3
29 Shamokin Drive
Don Mills, Ontario M3A 3H7
416-441-2417

Abstract

This paper describes a system of hardware and software which provides for the transfer of blocks of data between a VADCG Terminal Node Controller (TNC) and a CP/M system with a serial interface. Both the software to run in the TNC and in the CP/M system is included. The system provides block transfers, data transparency, flow control and error checking and retransmission in both directions over the interface.

Introduction

The software to implement the Link level of protocol for the VADCG Terminal Node Controller was developed in 1978. It is now in general use both in the U.S. and Canada and has even been implemented on other Terminal Node Controller boards. It has proven to be satisfactory for the purposes intended but many people recognize the need to implement the next higher level of protocol - the Packet or Network level protocol.

There have been a large number of proposals as to the form this protocol should take and I have made my own proposals in a paper published in the last Amateur Radio Computer Networking Conference. In spite of a large supply of proposals there is a distinct shortage of implementations. Part of the reason for this has been because of the need for some kind of consensus in the Amateur Radio fraternity. Notwithstanding this important concern, there is another reason why we don't have our Network level protocol implemented - it is a lot of work to get it going.

'What are the problems in implementing the Network level protocol?', you may ask. Well, unlike the Link level protocol which only had to be implemented to run in a TNC, parts of the Network level protocol have to be implemented to run in each microcomputer connected to the network. Furthermore, the TIP programs in the TNCs will have to be rewritten and some changes in the LIP programs are needed as well. In addition, the Network level protocol is much more complex than the link level protocol. I think one of the main stumbling blocks is the need to implement the protocol on two separate systems before any testing can be done.

Despite the above difficulties, I have begun the process of implementing the protocol and have broken the job down into steps that can be implemented and tested and then proceed to the next step. To alleviate the problem of having to make two implementations for different systems, I am only making one implementation for my CP/M system which I will hopefully be able to transport to another local Packeteer's CP/M system for testing. In order to make this program as transportable as possible to other CP/M systems I am only using the 8080 instruction set.

The programs here are not really any part of a higher level protocol but the function they perform will be needed by any higher level protocol that is adopted. The microcomputer program called 'PACKET' is basically a set of drivers for the serial interface between the microcomputer and the TNC. The program implementing the higher level of protocol in the microcomputer is called the Transmission Control Program or TCP. The TCP will use these drivers to transfer blocks of data that it has prepared to the TNC and it will also receive blocks of data from the TNC using these drivers.

The TCP is called upon by the programs running in the microcomputer to send data and receive data to and from various points in the network. In order to do this job, the TCP adds a header onto the outgoing blocks of data and because the bits and bytes in this header have a meaning based on their position in the block of data, there must be a mechanism to show where a block starts and ends in the serial data streams being passed accross the interface between the computer and the TNC. This mechanism, was lacking in all the TIPs that I had access to. Also, since flexibility in the setting of these bits was needed and any 'kind of restriction on the data being sent across the interface was undesirable, there had to be a mechanism for data transparency. This mechanism, too, was missing in all the TIPs that I had access to. Also, since data was being sent both ways at high speed by microprocessors, there had to be a mechanism for flow control in both directions across the interface. Also, since my serial interface used long RS-232 cables in a noisy environment, I occasionally got bit errors in the data especially at the higher speeds so I needed to have error detection in this interface. In some environments, error detection may not be necessary but I decided to play it safe and include it. Finally, error detection is not of much use unless you can correct the errors so I have incorporated a retransmission mechanism.

To summarize - the interface provides the following:

1. Block recognition.
2. Data transparency.
3. Flow control (in both directions)
4. Error detection.
5. Error correction.

A block has the following format:

SYN	DLE	STX	DATA	DLE	ETX	CRC	PAD
16H	10H	02H		10H	03H		FFH

The combination DLE-STX (ASCII Data Link Escape and Start of Text) indicates the start of a transparent block of data and the combination DLE-ETX indicates the end of the transparent block. To provide for data transparency a 'byte stuffing' technique is used - any time transparent data occurs that looks like a DLE, then an extra DLE is stuffed into the data stream. Therefore, the two byte combination DLE-DLE represents only a single data byte of 10H.

Flow control is accomplished using some hardware features of the TWC and the serial inter face on the microcomputer. The RTS (Request to Send) and CTS (Clear to Send) lines are cross connected and controlled by the programs. 'When the output line is high it means 'You can send data to me now'. When the output line is low it means 'Don't send any data to me now.'

Error detection is accomplished using the two-byte CRC (Cyclic Redundancy Check) characters following the FTX character in the block. I am using the following polynomial to generate the CRC bytes:

$$x^{16} + x^{15} + x^2 + 1$$

This is the usual polynomial used for synchronous protocols such as IBM BISYNC but is not the one suggested by the CCITT. On transmit, the CRC calculation is done on all transmitted characters after the STX and up to and including the ETX character. The stuffed bytes are included in

the calculation and after the STX is processed, two bytes of zeroes are processed. On receive, the calculation is the same except that the two CRC bytes are used instead of the zero bytes and the result of the CRC calculation will then come out to zero if everything was received correctly.

The error correction mechanism employed also utilizes some of the hardware features of the TNC and the microcomputer. The DTR (Data Terminal Ready) and the DSR (DataSet Ready) lines are cross connected between the TNC and microcomputer. Whenever one side receives a block correctly, it reverses the state of its output line. If the other side does not detect the transition then, after a timeout, it retransmits the block.

Hardware Requirements

In order to use the program called 'TIPTTC' which runs in the VADCG TNC, you will need a VADCG TNC with the serial interface installed and an RS232 cable with wires going to the following pins installed (2,3,4,5,6,7 and 20).

In order to use the program called 'PACKET' which runs in a CP/M system, you will need to have a serial interface capable of handling 8-bit characters, direct software control of two lines of RS-232 levels, and the ability to read two input RS-232 lines with the software. Most CP/M systems have this capability. It is true that I could have written this software to only require the data lines (and I may yet do this) and the software would be slightly more transportable but more complicated and a little less efficient. The flow control and acknowledgment systems work very well because the software in the TNC is alerted by the interrupt system almost instantly when there is any change in level of the interface lines.

Software Requirements

The 'TIPTTC' program should interface with any of the common LIP programs being used with the VADCG board. I can only think of one thing to watch out for - the program uses variables in the CCA (Common Communications Area) from displacement 40H to 54H so you should check your LIP's usage of these areas and relocate them if your LIP uses part of the same area. Also, make sure your stack does not get extended down as low as displacement 54H in the CCA. This is a 'vanilla' TIP and in addition to the features described above, it only has provision for connect and disconnect. If you use this TIP you will have to do without those special functions you previously had. The other alternative is to add the functions to this program yourself. If you take this latter option I would very much like to hear from you as well as anyone else who uses these programs. I like to get 'feedback.'

The 'PACKET' program only needs a CP/M system with the aforementioned hardware features and some configuration modifications described in the next section.

Confiauration Requirements

A. TIPTTC

A.1 At label 'BAUDRAT' the Baud rate may have to be changed. I am using 4800 Baud. In general it is best to have the rate as high as is reliable and convenient and should be 1200 or greater. However, lower Baud rates than 1200 would work as well.

A.2 At label 'ACKTO' there is a number which is related to the amount of time the TNC waits before retransmitting the block if no acknowledgment is received. This value has not been optimized from the first trial value. It is very non-critical and the value I chose for my system seems to work very well. It is probably quite a bit slower than required. You may experiment with different values.

A.3 At label 'RIMBUF' change the call sign to your own and if it is less than 6 characters, pad it on the right with blanks. Also, use upper case characters.

A.4 At label 'TERMNO' change your node number to whatever you want.

B. PACKET

B.1 In the section headed 'HARDWARE PORT EQUATES' you will have to change the port addresses to match the ports on your system.

B.2 In the sections headed 'CONTROL PORT BIT MEANINGS' and 'STATUS BIT MEANINGS' you will have to change the equates to match your system.

B.3 At label 'UARTINIT' change the code to initialize your serial interface UART to operate with 8 data bits and no parity bit. Also make the output lines going to pins 4 and 20 on the TNC are low. (The assumption here is that the jumper plug on the TNC is wired straight across)

B.4 At label 'SETRTS' change the code so that it makes pin 4 on the TNC end of the cable high.

B.5 At label 'CLEARRTS' change the code so that it makes pin 4 on the TNC end of the cable low.

B.6 At label 'FLIPDTR' make sure the code reverses the level on pin 20 of the TNC.

B.7 At label 'TESTTBE' test if data can be sent out to the UART and return non-zero status if it can.

B.8 At label 'TESTRDA' test if data is available from the UART and return non-zero status if it is.

B.9 At label 'TESTCTS' test the level of pin 5 coming from the TNC and return non-zero status if it is high.

B.10 At label 'TESTDSR' test the level of pin 6 coming from the TNC and compare it to the last tested level. If the value has changed, return non-zero status.

B.11 In routine 'KEYTEST' change the code to Look for a character to be entered on your keyboard and if there is none, then go to 'OUTTEST'. It will probably have to be changed because my keyboard uses inverted logic.

Operation

Although the importance of the "PACKET' program lies in the features provided by the drivers in it, I have added 25 instructions which allow the program to provide an immediately useful function. It will allow the user to use the keyboard and screen display in the CP/M system as if it were a terminal connected directly to the TNC. Because of the power of the driver code, it is a relatively trivial matter to add this function. Similarrily, a program to transfer a file from the system or to the system is very easy to implement using the drivers.

To use the program as a terminal simulator, simply type in a line of data on the keyboard, the line will be sent in a block to the TNC when the line feed key is pressed. While data is being entered after the first character, no blocks will be received from the TNC. While a block is being received from the TNC, the keyboard is not tested so a line that you enter will not be mixed with data coming from the TNC.

To connect, type the Call sign in upper case (which must be padded with blanks on the right if it is not 6 characters long) followed by control-A and then hit line feed to send it to the TNC. To disconnect, type any 6 characters (except for line feed) followed by control-B and then hit line feed. Sorry for this kludge but it is only temporary as I am planning to completely change the connect-disconnect procedures when I write the Transmission Control Program which is the next step in implementation of the Packet level protocols.

Summary

I hope these programs help those who are working on the implementation of the higher level protocols for an Amateur Radio digital communications network. It seemed to me that a program with these features would have to be one of the first steps in any kind of implementation but so far I have not heard of one. Perhaps someone out there has already written one and I have duplicated his effort. If so, then we are not doing enough advertising about what we have done. That is why I have taken this effort to disseminate the program.

The program listings here represent programs that have actually been running successfully so any problems encountered in transporting them to another system should be associated with the diffferent environment and not with defects in the code. I can supply the programs on standard SS-SD CP/M format diskettes if necessary. Please enclose $3.00 with a blank diskette or $8.00 without a diskette when making your request. You will find the listings for the two programs on the following pages.

* CP/M is a trade mark of Digital Research

```
                ********************************************************
                **     VADCG PACKET LEVEL TNC DRIVER FOR CP/M         **
                **  BY DOUG LOCKHART, VE7APU          JANUARY, 1983   **
                ********************************************************
                ; LAST CHANGED JANUARY 31, 1983

                ********************************************************
                * THIS PROGRAM CONTAINS THE DRIVERS TO EXCHANGE TRANS- *
                * PARENT BLOCKS OF DATA BETWEEN A CP/M OPERATING       *
                * SYSTEM AND A VADCG TERMINAL NODE CONTROLLER USING A  *
                * MATCHING  PROGRAM.  IT USES THE REQUEST TO SEND (RTS) *
                * AND CLEAR TO SEND (CTS) LINES FOR FLOW CONTROL AND   *
                * THE DATA SET READY (DSR) AND DATA TERMINAL READY     *
                * (DTR) LINES FOR ACKNOWLEDGEMENTS.  ONLY DATA INFOR-  *
                * MATION IS PASSED ON THE DATA LINES.  THE PROGRAM     *
                * USES 'BYTE STUFFING' TO ACHIEVE DATA TRANSPARENCY    *
                * AND USES A CRC-16 TO DETECT ERRORS.  IF THE TRANS-   *
                * MITTED DATA IS NOT ACKNOWLEDGED BY A CHANGE IN LEVEL *
                * THEN THE BLOCK IS SENT AGAIN.                        *
                ********************************************************
                        MISCELLANEOUS EQUATES
0005 =          LOS     EQU     5
                ;
                ;       ASCII EQUATES
000D =          CR      EQU     0DH     ; CARRIAGE RETURN
000A =          LF      EQU     0AH     ; LINE FEED
0010 =          DLE     EQU     10H     ; DATA LINK ESCAPE
0002 =          STX     EQU     02H     ; START OF TEXT
0003 =          ETX     EQU     03H     ; END OF TEXT
0016 =          SYN     EQU     16H     ; SYNCHRONIZING CHARACTER
00FF =          PAD     EQU     0FFH    ; PAD CHARACTER
                ********************************************************
                ;       HARDWARE PORT EQUATES
0001 =          DATA    EQU     01H             ; UART DATA PORT
0000 =          CONTROL EQU     00H             ; UART CONTROL PORT
0000 =          STATUS  EQU     00H             ; UART STATUS PORT
0002 =          KEYBD   EQU     02H             ; KEYBOARD DATA PORT
                ;
                ;       CONTROL PORT BIT MEANINGS
0001 =          DTR     EQU     01H             ; NOT DATA TERMINAL READY
0002 =          RTS     EQU     02H             ; NOT REQUEST TO SEND
0004 =          BRS0    EQU     04H             ; BAUD RATE SELECT
0008 =          BRS1    EQU     08H             ; BAUD RATE SELECT
0010 =          WLS1    EQU     10H             ; WORD LENGTH SELECT
0020 =          WLS2    EQU     20H             ; WORD LENGTH SELECT
0040 =          SBS     EQU     40H             ; STOP BIT SELECT
0080 =          PI      EQU     80H             ; PARITY INHIBIT
                ;
                ;       STATUS BIT MEANING
0001 =          RDA     EQU     01H             ; RECEIVE DATA AVAILABLE
0002 =          KSTB    EQU     02H             ; NOT KEYBOARD STROBE
0004 =          PE      EQU     04H             ; PARITY ERROR
0008 =          FE      EQU     08H             ; FRAMING ERROR
0010 =          OE      EQU     10H             ; OVERRUN ERROR
0020 =          DSR     EQU     20H             ; NOT DATA SET READY
0040 =          CTS     EQU     40H             ; NOT CLEAR TO SEND
0080 =          TBE     EQU     80H             ; TRANSMIT BUFFER EMPTY

                ********************************************************
0100                    ORG     100H
0100 31C903             LXX     SP,STACK        ; INITIALIZE STACK
0103 CD4301             CALL    UARTINIT        ; INITIALIZE UART
0106 CD3501     OUTTEST:CALL    WRITESTAT       ; ANY DATA IN TBUF?
0109 CA2301             JZ      LINETEST        ; NO, TRY TO RECEIVE SOME
010C DB00       KEYTEST:IN      STATUS          ; ANY KEYBOARD DATA?
010E E602               ANI     KSTB
0110 C20601             JNZ     OUTTEST         ; NO, TEST FOR LINE DATA
0113 DB02               IN      KEYBD           ; GET DATA
0115 CD3A01             CALL    DISPLAY         ; DISPLAY IT
0118 CD3A05             CALL    WRITE           ; PUT IT INTO BUFFER
011B FE0A               CPI     LF              ; WAS IT A LINE FEED?
011D CC1005             CZ      TCLOSE          ; YES,SEND DATA IN BUFFER
```

```
0120 C30601             JMP     OUTTEST         ; GO FOR MORE DATA
                LINETEST:
0123 CD3505             CALL    READSTAT        ; DATA IN RECEIVE BUFFER?
0126 CC3A04             CZ      BLOCKRX         ; NO, TRY TO RECEIVE SOME
0129 CA0C01             JZ      KEYTEST         ; NO, TEST KEYBOARD ENTRY
012C CD1805             CALL    READ            ; GET DATA BYTE FROM RBUF
012F CD3A01             CALL    DISPLAY         ; AND DISPLAY IT
0132 C32301             JMP     LINETEST
                WRITESTAT:
0135 3A9B02             LDA     TBUFNUM         ; GET COUNT
0138 B7                 ORA     A               ; AND TEST IT
0139 C9                 RET

013A F5         DISPLAY:PUSH    PSW
013B 5F                 MOV     E,A
013C 0E02               MVI     C,2
013E CD0500             CALL    BIOS            ; DISPLAY DATA IN (E)
0141 F1                 POP     PSW
0142 C9                 RET                     ; RETURN TO CALLER
                ********************************************************
                ;       BASIC UART DRIVER ROUTINES
                ;
                ;       INITIALZATION OF UART
                UARTINIT:
0143 3EB7               MVI     A,PI+WLS1+WLS2+BRS0+DTR+RTS     ; 8 DATA,
0145 D300               OUT     CONTROL ; NO PARITY, DTR AND RTS OFF
0147 329801             STA     CTRL            ; SAVE CONTROL INFO
014A DB01               IN      DATA            ; CLEAR ANY RESIDUAL DATA
014C DB00               IN      STATUS          ; SAVE INITIAL DSR STATUS
014E E620               ANI     DSR
0150 329701             STA     DSRSTAT
0153 C9                 RET                     ; RETURN TO CALLER

                ;       ENABLE RTS (MEANS DATA CAN BE RECEIVED)
0154 3A9801     SETRTS: LDA     CTRL            ; GET CONTROL INFORMATION
0157 E6FD               ANI     0FFH-RTS
0159 D300               OUT     CONTROL
015B 329801             STA     CTRL
015E C9                 RET

                ;       DISABLE RTS ( MEANS DO NOT SEND ME ANY DATA )
                CLEARRTS:
015F 3A9801             LDA     CTRL            ; GET CONTROL INFORMATION
0162 F602               ORI     RTS
0164 D300               OUT     CONTROL
0166 329801             STA     CTRL
0169 C9                 RET

                ;       REVERSE VALUE OF DTR (TO ACKNOWLEDGE BLOCK )
016A 3A9801     FLIPDTR:LDA     CTRL            ; GET CONTROL INFORMATION
016D EE01               XRI     DTR             ; FLIP DTR
016F D300               OUT     CONTROL
0171 329801             STA     CTRL    ; SAVE UART CONTROL INFORMATION
0174 C9                 RET                     ; RETURN TO CALLER

                ;       TEST VALUE OF TBE (TRANSMIT BUFFER EMPTY)
0175 DB00       TESTTBE:IN      STATUS
0177 E680               ANI     TBE
0179 C9                 RET

                ;       TEST IF RECIEVE DATA IS AVAILABLE
017A DB00       TESTRDA:IN      STATUS
017C E601               ANI     RDA
017E C9                 RET

                ;       TEST VALUE OF CLEAR TO SEND
                ;       NON-ZERO FLAG IF CTS, ZERO FLAG IF NO CTS
017F DB00       TESTCTS:IN      STATUS
0181 E640               ANI     CTS
0183 FE40               CPI     CTS             ; NOTE SENSE INVERTED
0185 C9                 RET
```

```
                ;       TEST IF VALUE OF DATA SET READY HAS CHANGED
                ;       NON-ZERO FLAG IF DSR HAS CHANGED, ZERO IF NOT
0186 E5         TESTDSR:PUSH    H               ; DO NOT CHANGE HL
0187 C5                 PUSH    B               ; OR BC
0188 219701             LX1     H,DSRSTAT       ; POINT AT OLD DSR STATUS
018B 46                 MOV     B,M
018C DB00               IN      STATUS
018E E620               AN1     DSR
0190 77                 MOV     M,A             ; SAVE NEW DSR STATUS
0191 B8                 CMP     B               ; COMPARE OLD AND NEW
0192 C1                 POP     B               ; RESTORE REGISTERS
0193 E1                 POP     H
0194 C9                 RET                     ; RETURN WITH FLAGS SET
                *****************************************************
0195 0000       CRC:    DW      0               ; CRC CALCULATION AREA
0197 00         DSRSTAT:DB      0               ; DSR STATUS SAVE AREA
0198 00         CTRL:   DB      0       ; CONTROL PORT INFORMATION
00FA =          MAXNUM  EQU     250     ; MAXIMUM AMOUNT OF DATA ALLOWED
0199 9C01       RPOINT: DW      RBUF    ; NEXT POINT TO GET DATA IN RBUF
019B 00         RBUFNUM:DB      0       ; NUMBER OF BYTES IN RBUF
019C            RBUF:   DS      253             ; RECEIVE BUFFER
0299 9C02       TPOINT: DW      TBUF    ; NEXT POINT TO PUT DATA TN TBUF
029B 00         TBUFNUM:DB      0       ; NUMBER OF BYTES TN TBUF
029C            TBUF:   DS      253             ; TRANSMIT BUFFER
0399                    DS      30H             ; STACK AREA
03C9 =          STACK   EQU     $
                *****************************************************
                ;       SEND BYTE OF DATA OUT TO SERIAL PORT
                ;       DATA PASSED IN ACCUMULATOR

                SENDDATA:
03C9 CD5205             CALL    CALCCRC         ; INCLUDE IN CRC
03CC E5         SEND:   PUSH    H
03CD C5                 PUSH    B               ; SAVE B&C
03CE 4F         SEND1:  MOV     C,A             ; SAVE DATA TEMPORARILY
03CF 210100             LX1     H,1             ; DELAY FOR BUG IN UART
03D2 2B         SEND2:  DCX     H
03D3 7C                 MOV     A,H
03D4 B5                 ORA     L               ; IS IT o?
03D5 C2D203             JNZ     SEND2
03D8 CD7501     SEND3:  CALL    TESTTBE         ; IS TBUF EMPTY?
03DB CAD803             JZ      SEND3   ; NO, KEEP LOOPING UNTIL IT TS.
03DE CD7F01     SEND4:  CALL    TESTCTS         ; IS CLEAR TO SEND UP?
03E1 CADE03             JZ      SEND4           ; NO, CAN'T SEND YET
03E4 79                 MOV     A,C             ; GET BACK DATA BYTE
03E5 D301               OUT     DATA
03E7 C1                 POP     B               ; RESTORE B
03E8 E1                 POP     H
03E9 C9                 RET                     ; RETURN TO CALLER

                ;       SEND DATA IN TBUF TO THE UART TRANSPARENTLY

                SENDTBUF:
03EA E5                 PUSH    H
03EB C5                 PUSH    B
03EC 219B02             LX1     H,TBUFNUM       ; POINT TO TBUF BYTE CNT
03EF 4E                 MOV     C,M             ; SAVE IN C
                SENDTBUF1:
03F0 23                 INX     H               ; POINT TO NEXT BYTE
03F1 7E                 MOV     A,M             ; GET IT
03F2 CDC903             CALL    SENDDATA        ; SEND IT
03F5 FE10               CPI     DLE             ; WAS IT DLE?
03F7 CCC903             CZ      SENDDATA        ; IF SO, SEND IT AGAIN
03FA 0D                 DCR     C               ; DECREMENT COUNT
03FB C2F003             JNZ     SENDTBUF?       ; CONTINUE SENDING
03FE C1                 POP     B
03FF E1                 POP     H
0400 C9                 RET                     ; RETURN TO CALLER

                  SEND FORMATTED BLOCK TO UART
                BLOCKTX:
0401 3E16               MVI     A,SYN
0403 CDCC03             CALL    SEND
0406 3E10               MVI     A,DLE
0408 CDCC03             CALL    SEND
040B 3E02               MVI     A,STX
040D CDCC03             CALL    SEND
0410 210000             LX1     H,0             ; INITIALIZE CRC AREA
0413 229501             SHLD    CRC
0416 CDEA03             CALL    SENDTBUF        ; SEND DATA IN TBUF
0419 3E10               MVI     A,DLE           ; THEN DLE-ETC SEQUENCE
041B CDC903             CALL    SENDDATA        ; INCLUDE IN CRC
041E 3E03               MVI     A,ETX
0420 CDC903             CALL    SENDDATA        ; INCLUDE IN CRC
0423 CDDE04             CALL    SENDCRC         ; FINALLY SEND CRC BYTES
0426 CDF204             CALL    CHECKRX         ; TRY TO RECEIVE
0429 CDD004             CALL    WAITDSR         ; WAIT FOR DSR TO CHANGE
042C CA0104             JZ      BLOCKTX ; DIDN'T CHANGE, SEND BLOCK AGAIN
042F AF                 XRA     A               ; A <-- 0
0430 329B02             STA     TBUFNUM         ; INDICATE TBUF IS EMPTY
0433 219C02             LX1     H,TBUF          ; POINT TO START OF TBUF
0436 229902             SHLD    TPOINT
0439 C9                 RET                     ; RETURN TO CALLER

                ;       READ A FORMATTED TRANSPARENT BLOCK OF DATA
043A CD5401     BLOCKRX:CALL    SETRTS          ; ALLOW OTHER END TO SEND
                BLOCKRX1:
043D CDAE04             CALL    RECEIVE         ; READ A BYTE FROM LINE
0440 C8                 RZ      ; RETURN WITH ZERO STATUS IF TIMED OUT
0441 FE10               CPI     DLE             ; IS IT DLE?
0443 C23D04             JNZ     BLOCKRX1        ; NO, KEEP TRYING
0446 CDAE04             CALL    RECEIVE         ; GOT DLE, TRY FOR STX
0449 C8                 RZ      ; RETURN WITH ZERO STATUS IF TIMED OUT
044A FE02               CPI     STX             ; IS IT STX?
044C C23D04             JNZ     BLOCKRX1        ; NO, TRY FOR DLE AGAIN
                BLOCKRX2:                       ; ENTRY FROM BLOCKTX
044F 219C01             LX1     H,RBUF          ; POINT TO START OF RBUF
0452 229901             SHLD    RPOINT
0455 CD8004             CALL    RCVRBUF         ; RECEIVE DATA INTO RBUF
                                ; UNTIL A CONTROL SEQUENCE IS RECEIVED
0458 C8                 RZ      ; RETURN ZERO STATUS IF LINE TIMES OUT
0459 FE03               CPI     ETX             ; WAS IT ETX?
045B C23D04             JNZ     BLOCKRX1        ; UNEXPECTED SEQUENCE
045E CDAE04             CALL    RECEIVE         ; RECEIVE FIRST CRC CHAR
0461 C8                 RZ                      ; RETURN HERE IF TIME OUT
0462 CDAE04             CALL    RECEIVE         ; RECEIVE SECOND CRC CHAR
0465 C8                 RZ                      ; RETURN HERE IF TIME OUT
0466 CD5F01             CALL    CLEARRTS        ; STOP OTHER END
0469 2A9501             LHLD    CRC             ; CHECK IF CRC WAS OK
046C 7C                 MOV     A,H
046D B5                 ORA     L
046E C27C04             JNZ     BLOCKRX3        ; NO GOOD
0471 219B01             LX1     H,RBUFNUM       ; SAVE DATA COUNT
0474 71                 MOV     M,C
0475 CD6A01             CALL    FLIPDTR         ; GOOD, REVERSE DTR LINE
                                                ; TO ACKNOWLEDGE BLOCK
0478 3EFF               MVI     A,-1            ; RETURN NON-ZERO STATUS
047A B7                 ORA     A               ; BLOCK RECEIVED OK
047B C9                 RET                     ; RETURN TO CALLER
                BLOCKRX3:
047C 3E00               MVI     A,0             ; RETURN WITH ZERO STATUS
047E B7                 ORA     A               ; NO BLOCK RECEIVED
047F C9                 RET


                ;       RECEIVE DATA PORTION OF BLOCK, RETTJRNS WHEN A
                ;       CONTROL SEQUENCE FOUND IN THE TRANSPARENT TEXT
0480 210000     RCVRBUF:LXI     H,0             ; INITIALIZE CRC TO 0
0483 229501             SHLD    CRC
0486 219C01             LXI     H,RBUF          ; POINT TO START OF RBUF
0489 0E00               MVI     C,0             ; BYTE COUNT = 0
                RCVRBUF1:
048B CDAE04             CALL    RECEIVE         ; GET A BYTE FROM LINE
048E C8                 Rz      ; RETURN HERE WITH ZERO STATUS IF TIMEOUT
```

```
048F FE10              CPI     DLE             ; WAS IT DLE?
0491 CA9A04            JZ      RCVRBUF3        ; YES, LOOK AT NEXT BYTE
               RCVRBUF2:
0494 77                MOV     M,A             ; PUT INTO BUFFER
0495 23                INX     H               ; INCREMENT RBUF POINTER
0496 0C                INR     C               ; INCREMENT COUNT
0497 C38B04            JMP     RCVRBUF1        ; LOOP FOR NEXT BYTE
               RCVRBUF3:
049A CDAE04            CALL    RECEIVE
049D C8                RZ      ; ZERO STATUS RETURN IF LINE TIMES OUT
049E FE10              CPI     DLE     ; IS IT A TRANSPARENT DLE?
04A0 CA9404            JZ      RCVRBUF2        ; YES, GO PUT INTO BUFFER
04A3 C9                RET     ; RETURN WITH CONTROL BYTE IN ACCUMULATOR
                                               ; AND NON-ZERO STATUS

               ;       TRY TO READ PROM LINE WITH LONG TIMEOUT
04A4 E5        RECEIVL:PUSH    H
04A5 CD5401            CALL    SETRTS          ; ALLOW OTHER END TO SEND
04A8 21A00F            LXI     H,4000          ; LONG TIMEOUT VALUE
                               ; SHOULD BE ADJUSTED FOR BEST RESULTS
04AB C3B504            JMP     RECEIV1

               ;       TRY TO READ FROM LINE, IF LINE TIMES OUT,
               ;       RETURN WITH ZERO STATUS
04AE E5        RECEIVE:PUSH    H
04AF CD5401            CALL    SETRTS          ; ALLOW OTHER END TO SEND
04B2 210007            LX1     H,2000          ; SHORT TIMEOUT VALUE,
                               ; ADJUST FOR ABOUT 2 CHAR TIMES
04B5 CD7A01    RECEIV1:CALL    TESTRDA         ; ANY RECEIVED DATA?
0488 CAC204            JZ      RECEIV2         ; NO, DECREMENT TIME
04BB DBO?              IN      DATA            ; GET DATA BYTE
04BD CD5205            CALL    CALCCRC         ; INCLUDE IT IN CRC
04C0 El                POP     H
04C1 C9                RET     ; GOOD RETURN WITH NON ZERO STATUS
04C2 2B        RECEIV2:DCX     H               ; DECREMENT TIMER
04C3 7C                MOV     A,H
04C4 B5                ORA     L               ; IS TIME OVER
04C5 C2B504            JNZ     RECEIV1         ; NO, KEEP TRYING
04C8 CD5F01            CALL    CLEARRTS        ; OOPS, TIMED OUT,
                               ; DROP RTS SO OTHER SIDE WILL STOP
04CB 3E00              MVI     A,0     ; RETURN WITH ZERO STATUS
04CD B7                ORA     A
04CE E1                POP     H
04CF C9                RET

               ;       WAIT FOR DSR TO CHANGE USING TIMEOUT
               WAITDSR:
04D0 011027            LXI     B, 10000        ; DELAY - ALTER AS REQ'D
               WAITDSR?:
04D3 CD8601            CALL    TESTDSR         ; CHECK FOR DSR CHANGE
04D6 CO                RNZ                     ; RETURN IF IT HAS
04D7 0B                DCX     B
04D8 78                MOV     A,B
04D9 B1                ORA     C               ; IS TIME OVER?
04DA C2D304            JNZ     WAITDSR1        ; NO CONTINUE TESTING
04DD C9                RET                     ; UNSUCCESSFUL RETURN

               ;       SENDTHECRC  BYTES
04DE AF        SENDCRC:XRA     A               ; FINISH CRC CALCULATION
04DF CD5205            CALL    CALCCRC
04E2 CD5205            CALL    CALCCRC
04E5 3A9601            LDA     CRC+1           ; SEND FIRST CRC CHAR
04E8 CDCC03            CALL    SEND
04EB 3A9501            LDA     CRC             ; SEND SECOND CRC CHAR
04EE CDCC03            CALL    SEND
04F1 C9                REX

               ;       CHECK IF WE CAN RECEIVE A BLOCK NOW
               ;       THIS ROUTINE IS CALLED AFTER A BLOCK HAS BEEN
               ;       TRANSMITTED TOALLOW THE OTHER SIDE TO GET A
               ;       CHANCE TO SEND TO US
04F2 3A9B01    CHECKRX: LDA    RBUFNUM         ; IS THERE ANY DATA LEFT
```

```
04F5 B7                ORA     A               ; IN RECEIVE BUFFER?
04F6 C0                RNZ                     ; YES, CAN'T RECEIVE
04F7 CD5401            CALL    SETRTS          ; ENABLE RTS TO ALLOW
                                               ; OTHER SIDE TO SEND
               CHECKRX1:
04FA CDA404            CALL    RECEIVL         ; READ WITH LONG TIMEOUT
04FD C8                RZ                      ; TIMED OUT , RETURN
04FE FE10              CPI     DLE             ; IS IT A DLE?
0500 C2FA04            JNZ     CHECKRX1        ; NO, KEEP LOOKING
0503 CDA404            CALL    RECEIVL         ; NOW LOOK FOR A STX
0506 C8                RZ                      ; TIMED OUT SO RETURN
0507 FE02              CPI     STX             ; IS IT START OF TEXT?
0509 C2FA04            JNZ     CHECKRX1        ; NO, KEEP LOOKING
050C CD4F04            CALL    BLOCKRX2        ; NOW GO READ TRANSP.TEXT
050F C9                RET                     ; ZERO STATUS IF TIMEOUT
                                       ; NON-ZERO IF BLOCK WAS RECEIVED

               ;       SEND A BLOCK OF TRANSMIT DATA TO THE LINE I?
               ;       THERE IS ANY DATA IN THE BUFFER
0510 3A9B02    TCLOSE: LDA     TBUFNUM         ; GET COUNT IN BUFFER
0513 B7                ORA     A               ; TEST FOR DATA
0514 C40104            CNZ     BLOCKTX         ; SEND BLOCK IF ANY DATA
0517 C9                RET                     ; RETURN TO CALLER

0518 ES        READ:   PUSH    H               ; SAVE HL
0519 219B01    READ1:  LXI     H,RBUFNUM       ; POINT AT COUNT IN RBUF
051C 7E                MOV     A,M
051D B7                ORA     A               ; IS THERE MY LEFT?
051E C22A05            JNZ     READ2           ; YES
0521 CD1005            CALL    TCLOSE          ; SEND ANY DATA IN TBUF
0524 CD3A04            CALL    BLOCKRX         ; RECEIVE ANOTHER BLOCK
0527 C31905            JMP     READ1           ; TRY TO DO READ AGAIN
052A 35        READ2:  DCR     M               ; DECREMENT COUNT
052B 2A9901            LHLD    RPOINT          ; GET READ POINTER
052E 7E                MOV     A,M             ; GET DATA BYTE
052F 23                INX     H               ; INCREMENT POINTER
0530 229901            SHLD    RPOINT          ; AND SAVE AGAIN
0533 El                POP     H               ; RESTORE HL
0534 C9                RET             ; RETURN TO CALLER WITH DATA IN A

               READSTAT:
0535 3A9B01            LDA     RBUFNUM ; GET COUNT OF DATA IN BUFFER
0538 B7                ORA     A               ; TEST IT
0539 C9                RET             ; NON-ZERO STATUS IF DATA PRESENT

053A F5        WRITE:  PUSH    PSW             ; SAVE DATA
053B E5                PUSH    H               ; SAVE HL
053C 2A9902            LHLD    TPOINT          ; GET POINTER INTO TBUF
053F 77                MOV     M,A             ; PUT DATA INTO BUFFER
0540 23                INX     H               ; INCREMENT POINTER
0541 229902            SHLD    TPOINT
0544 219B02            LX1     H,TBUFNUM       ; POINT To COUNT IN TBUF
0547 7E                MOV     A,M             ; INCREMENT COUNT
0548 3C                INR     A
0549 77                MOV     M,A
054A FEFA              CPI     MAXNUM          ; IS BUFFER FULL?
054C cc0104            CZ      BLOCKTX         ; YES, SEND BLOCK NOW
054F E?                POP     H               ; RESTORE HL
0550 F1                POP     PSW             ; RESTORE  DATA
0551 C9                RET

               ; CRC CALCULATION ROUTINE
               ; USES BYTE PASSED IN ACCUMULATOR TO INCLUDE IN CRC
               ; RESTORES ALL REGISTERS AND STATUS
0552 E5        CALCCRC:PUSH    H
0553 c5                PUSH    B
0554 F5                PUSH    PSW
0555 0608              MVI     B,8
0557 4F                MOV     C,A
0558 2A9501            LHLD    CRC
055B =         CALCCRC1:EQU    $
055B 79                MOV     A,C
```

```
055C 07                  RLC
055D 4F                  MOV    C,A
055E 7D                  MOV    A,L
055F 17                  RAL
0560 6F                  MOV    L,A
0561 7C                  MOV    A,H
0562 17                  RAL
0563 67                  MOV    H,A
0564 D26F05              JNC    CALCCRC2
0567 7C                  MOV    A,H
0568 EE80                XRI    80H
056A 67                  MOV    H,A
056B 7D                  MOV    A,L
056C EE05                XRI    05H
056E 6F                  MOV    L,A
                CALCCRC2:
056F 05                  DCR    B
0570 C25B05              JNZ    CALCCRC1
0573 229501              SHLD   CRC
0576 F1                  POP    PSW
0577 C1                  POP    B
0578 E1                  POP    H
0579 C9                  RET

057A                     END
```



```
        ; ***********************************************************
        ; **    VADCG TERMINAL NODE COMMUNICATIONS PROGRAM        **
        ; **    BY DOUG LOCKHART, VE7APU      JANUARY, 1983       **
        ; ***********************************************************
        ; LAST CHANGED: JANUARY 31, 1983

        ; TERMINAL INTERFACE PROGRAM FOR INTERFACING TO A CP/M
        ; SYSTEM.  THIS PROGRAM IS WRITTEN TO RUN IN THE VADCG
        ; TERMINAL NODE CONTROLLER.  IT INTERFACES WITH A LINK
        ; INTERFACE PROGRAM (LIP) RUNNING AT ADDRESS 0 IN MEMORY.
        ; THIS VERSION IS WRITTEN TO USE THE 8250 PROGRAMMABLE
        ; UART TO COMMUNICATE WITH A COMPUTER.
        ; THE BASIC FEATURES OF THIS TIP ARE:
        ; TRANSFER OF DATA IN BLOCKS
        ; RTS FLOW CONTROL FROM DIGITAL EQUIPMENT TO TIP
        ; AND CTS FLOW CONTROL FROM TIP TO DIGITAL EQUIPMENT
        ; ACKNOWLEDGEMENTS TO BLOCKS RECEIVED BY A CHANGE IN DTR
        ; ACKNOWLEDGEMENTS TO BLOCKS SENT BY A CHANGE IN DSR
        ; CRC-16 CHECKING OF ALL DATA BLOCKS
        ; ERROR RECOVERY BY RETRANSMISSION IF NO ACKNOWLEDGMENT
        ; USES BYTE STUFFING TECHNIQUE FOR DATA TRANSPARENCY

        INCTB   MACRO   ?D
                IF      NOT NUL ?D
                MVI     A,?D
                ENDIF
                RST     2
                ENDM

        INCLB   MACRO   ?D
                IF      NOT NUL ?D
                MVI     A,?D
                ENDIF
                RST     3
                ENDM

        COMPARE MACRO
                RST     5
                ENDM

        SIM     MACRO
                DB      30H     ; SET INTERRUPT MASK
                ENDM

        RIM     MACRO
                DB      20H     ; READ INTERRUPT MASK
                ENDM


        ; RAM CONSTANT - CHANGE FOR DIFFERENT RAM LOCATION
1000 =  LORAM   EQU     1000H   ; START OF RAM STORAGE


        ; NON-ZERO STATUS MEANS LINE BUFFER ADDRESS IS IN HL REG.
        ; ZERO STATUS MEANS NO BUFFER IS READY
        NEXTIN  MACRO
                RST     4
                ENDM


        ; 8255 PARALLEL I/O EQUATES

0008 =  PORTA   EQU     8       ; PORT A INPUT AND OUTPUT
0009 =  PORTB   EQU     9       ; PORT B INPUT AND OUTPUT
000A =  PORTC   EQU     0AH     ; PORT C INPUT AND OUTPUT
000B =  CONTROL EQU     0BH     ; CONTROL PORT OUTPUT ONLY


        ; BAUD RATE EQUATES
0004 =  BAUD384 EQU     4       ; DIVISOR FOR 38,400 BAUD
0008 =  BAUD192 EQU     8       ; DIVISOR FOR 19,200 BAUD
0010 =  BAUD96  EQU     16      ; DIVISOR FOR 9600 BAUD
```

```
0020 =          BAUD48  EQU     32      ; DIVISOR FOR 4800 BAUD
0040 =          BAUD24  EQU     64      ; DIVISOR FOR 2400 BAUD
0080 =          BAUD12  EQU     128     ; DIVISOR FOR 1200 BAUD
0100 =          BAUD600 EQU     256     ; DIVISOR FOR 600 BAUD
0200 =          BAUD300 EQU     512     ; DIVISOR FOR 300 BAUD
0400 =          BAUD150 EQU     1024    ; DIVISOR FOR 150 BAUD
0476 =          BAUD134 EQU     1142    ; DIVISOR FOR 134.5 BAUD
0573 =          BAUD110 EQU     1395    ; DIVISOR FOR 110 BAUD
0800 =          BAUD75  EQU     2048    ; DIVISOR FOR 75 BAUD
0C00 =          BAUD50  EQU     3072    ; DIVISOR FOR 50 BAUD


                ; 8250 SERIAL I/O EQUATES

                ; REGISTER EQUATES
0000 =          RBR     EQU     0       ; RECEIVE BUFFER REGISTER  (R)
0000 =          THR     EQU     0       ; TRANSMIT HOLDING REGISTER (W)
0001 =          IER     EQU     1       ; INTERRUPT ENABLE REGISTER  (W)
0002 =          IIR     EQU     2       ; INTERRUPT IDENT. REGISTER (R)
0003 =          LCR     EQU     3       ; LINE CONTROL REGISTER (R/W)
0004 =          MCR     EQU     4       ; MODEM CONTROL REGISTER (R/W)
0005 =          LSR     EQU     5       ; LINE STATUS REGISTER (R/W)
0006 =          MSR     EQU     6       ; MODEM STATUS REGISTER (R/W)
0000 =          DLL     EQU     0       ; DRIVER LATCH (LSB) (W)
0001 =          DLM     EQU     1       ; DRIVER LATCH (MSB) (W)

                ; INTERRUPT ENABLE EQUATES
0001 =          ERBFI   EQU     1       ; ENABLE RECEIVED DATA INTERRUPT
0002 =          ETBEI   EQU     2       ; ENABLE TRANSMITTER
0004 =          ELSI    EQU     4       ; RECEIVER LINE STATUS INTERRUPT
0008 =          EDSSI   EQU     8       ; ENABLE MODEM STATUS INTERRUPT

                ; INTERRUPT IDENTIFICATION EQUATES
0001 =          IPEND   EQU     1       ; '0' IF INTERRUPT PENDING
0002 =          IID0    EQU     2       ; INTERRUPT IDENTIFICATION BIT 0
0004 =          IID1    EQU     4       ; INTERRUPT IDENTIFICATION BIT 1

                ; LINE CONTROL EQUATES
0001 =          WLS0    EQU     1       ; WORD LENGTH SELECT BIT 0
0002 =          WLS1    EQU     2       ; WORD LENGTH SELECT BIT 1
0004 =          STB     EQU     4       ; STOP BIT SELECT
0008 =          PEN     EQU     8       ; PARITY ENABLE
0010 =          EPS     EQU     10H     ; EVEN PARITY SELECT
0020 =          SPTY    EQU     20H     ; STICK PARITY
0040 =          SBRK    EQU     40H     ; SET BREAK
0080 =          DLAB    EQU     80H     ; DRIVER LATCH ACCESS BIT

                ; MODEM CONTROL EQUATES
0001 =          DTR     EQU     1       ; DATA TERMINAL READY
0002 =          RTS     EQU     2       ; REQUEST TO SEND
0004 =          OUT1    EQU     4       ; OUT1 LINE ON 8250
0008 =          OUT2    EQU     8       ; OUT2 LINE ON 8250
0010 =          LOOP    EQU     10H     ; MODEM LOOP CONTROL BIT

                ; LINE STATUS EQUATES
0001 =          DR      EQU     1       ; DATA READY
0002 =          OE      EQU     2       ; OVERRUN ERROR
0004 =          PE      EQU     4       ; PARITY ERROR
0008 =          FE      EQU     8       ; FRAMING ERROR
0010 =          BI      EQU     10H     ; BREAK INTERRUPT
0020 =          THRE    EQU     20H     ; TRANSMITTER HOLDING REG EMPTY
0040 =          TSRE    EQU     40H     ; TRANSMITTER SHIFT REG EMPTY

                ; MODEM STATUS EQUATES
0001 =          DCTS    EQU     1       ; DELTA CLEAR TO SEND
0002 =          DDSR    EQU     2       ; DELTA DATA SET READY
0004 =          TERI    EQU     4       ; TRAILING EDGE RING INDICATOR
0008 =          DRLSD   EQU     8       ; DELTA RX LINE SIGNAL DETECT
0010 =          CTS     EQU     10H     ; CLEAR TO SEND
0020 =          DSR     EQU     20H     ; DATA SET READY
0040 =          RI      EQU     40H     ; RING INDICATE
```

```
0080 =          RLSD    EQU     80H     ; RECEIVE LINE SIGNAL DETECT

0017 =          RIMD    EQU     17H     ; REQUEST INITIALIZATION MODE
0008 =          MSE     EQU     08H     ; MASK SET ENABLE BIT


                ; COMMON COMMUNICATIONS AREA

                ; CIRCULAR TERMINAL BUFFER VARIABLES

1000 =          CCA     EQU     LORAM   ; COMMON COMMUNICATIONS AREA ADR.
1004 =          CTBIE   EQU     CCA+4   ; CURRENT TERMINAL BUF INP. ENTRY
1006 =          OTBE    EQU     CCA+6   ; OLDEST TERMINAL BUFFER ENTRY
1008 =          TBIP    EQU     CCA+8   ; TERMINAL BUFFER INPUT POINTER
100A =          TBOP    EQU     CCA+0AH ; TERMINAL BUFFER OUTPUT POINTER
100C =          LTBOE   EQU     CCA+0CH ; LAST TERMINAL BUF OUTPUT ENTRY
100E =          CTBOE   EQU     CCA+0EH ; CURRENT TERMINAL BUF OUT ENTRY

                ; CIRCULAR LINE BUFFER VARIABLES

1012 =          LBPE    EQU     CCA+12H ; LINE BUFFER PROCESSING ENTRY
1014 =          CLBE    EQU     CCA+14H ; CURRENT LINE BUFFER ENTRY ADDR.
1016 =          OLBE    EQU     CCA+16H ; OLDEST LINE BUFFER ENTRY
1018 =          LBIP    EQU     CCA+18H ; LINE BUFFER INPUT POINTER
101A =          LBOP    EQU     CCA+1AH ; LINE BUFFER OUTPUT POINTER

                ; MISCELLANEOUS


1000 =          STAT1   EQU     CCA     ; MAINLINE STATUS BYTE

                ; THE FOLLOWING VARIABLES ARE FOR EXCLUSIVE USE BY TIP

101C =          BUFCOUNT EQU    CCA+1CH ; CURRENT INPUT BUFFER COUNT
101D =          OUTCOUNT EQU    CCA+1DH ; CURRENT OUTPUT BYTES REMAINING
1040 =          WAIT    EQU     CCA+40H ; CHARACTER DELAY VALUE
1042 =          MSRSAVE EQU     CCA+42H ; LATEST MODEM STATUS REGISTER
1043 =          INTFLAG EQU     CCA+43H ; INTERRUPT ROUTINE FLAGS
0001 =          RXBUSY  EQU     01H     ; RECEIVE INTRPT ROUTINE ACTIVE
0002 =          TXBUSY  EQU     02H     ; TRANSMIT INTRPT ROUTINE ACTIVE
1044 =          CRC     EQU     CCA+44H ; CRC CALCULATION AREA
1046 =          RCRC2   EQU     CCA+46H ; SECOND RECEIVED CRC BYTE
1047 =          RCRC1   EQU     CCA+47H ; FIRST  RECEIVED CRC BYTE
1048 =          TCRC2   EQU     CCA+48H ; SECOND TRANSMIT CRC BYTE
1049 =          TCRC1   EQU     CCA+49H ; FIRST TRANSMIT CRC BYTE
104A =          RNEXT   EQU     CCA+4AH ; CURRENT RECEIVE ROUTINE ADDRESS
104C =          TNEXT   EQU     CCA+4CH ; TRANSMIT ROUTINE ADDRESS
104E =          RDISP   EQU     CCA+4EH ; RECEIVE INTERRUPT ROUTINE ADDR.
1050 =          TDISP   EQU     CCA+50H ; TRANSMIT INTERRUPT ROUTINE ADDR
1052 =          DFLAG   EQU     CCA+52H ; DISPATCH FLAG
0001 =          CRCTX   EQU     01H     ; CRC ROUTINE IN USE BY TX DISP.

                ; ASCII EQUATES
000D =          CR      EQU     0DH     ; ASCII CARRIAGE RETURN
000A =          LF      EQU     0AH     ; ASCII LINE FEED
001B =          ESC     EQU     1BH     ; ASCII ESCAPE CHARACTER
0002 =          STX     EQU     02H     ; ASCII START OF TEXT
0003 =          ETX     EQU     03H     ; ASCII END OF TEXT
0010 =          DLE     EQU     10H     ; ASCII DATA LINK ESCAPE
0016 =          SYN     EQU     16H     ; ASCII SYNCHRONIZATION CHARACTER
00FF =          PAD     EQU     0FFH    ; TRAILING PAD CHARACTER

00FF =          TRUE    EQU     0FFH    ; FOR IF CONDITION TESTS
0000 =          FALSE   EQU     0       ; FOR IF CONDITION TESTS

                ; ************************************************************
                ; **              CONFIGURATION EQUATES                     **
                ; **       VALUES CHANGE FOR EVERY CONFIGURATION            **
                ; ************************************************************


0003 =          FORMAT  EQU     WLS1+WLS0       ; UART FORMAT (8 DATA,
```

```
                                         ; NO PARITY)
0020 =           BAUDRAT EQU     BAUD48  ; CURRENT BAUD RATE

00FF =           CUSHION EQU     255     ; THE MINIMUM NUMBER OF BYTES
                                 ; AVAILABLE IN THE TERMINAL BUFFER THAT
                                 ; ARE REQUIRED BEFORE A RECEIVE
                                 ; OPERATION IS STARTED.

2710 =           ACKTO   EQU     10000   ; ACKNOWLEDGE TIMEOUT COUNT
                                         ; (PRELIMINARY VALUE)

                 *********************************************************
0800                     ORG     800H    ; THIS PROGRAMS EPROM START ADR.

                 ; ENTRY JUMP TABLE

0800 C31508              JMP     TJPINIT ; INITIALIZATION ENTRY POINT
0803 C34808              JMP     RST55   ; INTERRUPT FROM 8250
0806 C30608              JMP     $       ; UNUSED INTERRUPT ENTRY POINT
0809 C3100A              JMP     DISPRX  ; TO DISPATCHER ROUTINE
080C 0C17564537RIMBUF    DB      12,RIMD,'VE7APU'        ; CONNECT BUFFER
0814 C8          TERMNO  DB      200     ; THIS NODES TERMINAL NUMBER
                 *********************************************************
                 TIPINIT:
                         ; SET BAUD RATE IN SERIAL PORT
0815 3E80                MVI     A,DLAB
0817 D303                OUT     LCR
0819 3E20                MVI     A,LOW BAUDRAT
081B D300                OUT     DLL             ; BAUD RATE DIVISOR LSB
081D 3E00                MVI     A,HIGH BAUDRAT
081F D301                OUT     DLM             ; BAUD RATE DIVISOR MSB

                         ; DEFINE CHARACTER FORMAT OF SERIAL DATA
0821 3E03                MVI     A,FORMAT
0823 D303                OUT     LCR     ; UPDATE LINE CONTROL REGISTER

                         ; UNMASK INTERRUPTS FROM SERIAL INTERFACE
                         RIM             ; GET CURRENT INTERRUPT MASK IN A
0825+20                  DB      20H     ; READ INTERRUPT MASK
0826 E606                AN1     00000110B       ; RESET RST5.5 MASK BIT
0828 F608                OR1     MSE     ; SET MASK SET ENABLE BIT
                         SIM             ; ENABLE RST5.5 INTERRUPTS
082A+30                  DB      30H     ; SET INTERRUPT MASK

                         ; CLEAR OUT RECEIVE BUFFER REGISTER
082B DB00                IN      RBR

                         ; SET UP INITIAL DISPATCH ROUTINES
082D 214309              LX1     H,EXIT  ; SET RECEIVE INTERRUPT TO IDLE
0830 224A10              SHLD    RNEXT
0833 211608              LX1     H,WAITLIP       ; WAITING FOR LIP BLOCK
0836 225010              SHLD    TDISP
0839 21140A              LXI     H,WAITTB        ; WAITING FOR FREE CUSHION
083C 224310              SHLD    RDISP
                 ; ENABLE RECEIVED DATA AVAILABLE AND MODEM STATUS INTRPT
083F 3E09                MVI     A,ERBFI+EDSSI   ; RECEIVE AND MODEM
0841 D301                OUT     IER     ; UPDATE INTERRUPT REGISTER

                         ; BRING UP RLSD (OUT1 = RLSD)
0843 3E04                MVI     A,OUT1
0845 D304                OUT     MCR     ; UPDATE MODEM CONTROL REGISTER

                         ; RETURN TO LIP FOR COMPLETION OF INITIALIZATION
0847 C9                  RET
                 *********************************************************
0848 F5          RST55:  PUSH    PSW
0849 E5                  PUSH    H
084A D5                  PUSH    D
084B C5                  PUSH    B
084C DB02                IN      IIR     ; GET INTERRUPT IDENT INFORMATION
084E FE04                CPI     IID1    ; RECEIVED DATA AVAILABLE INTRPT?
0850 CA8A08              JZ      RXINT   ; GO TO RECEIVE INTERRUPT ROUTINE
```

```
0853 FE02                CPI     1100    ; IS IT TRANSMIT BUFFER EMPTY
0855 CA4909              JZ      TXINT   ; GO TO TRANSMIT INTRPT ROUTINE
0858 B7                  ORA     A       ; MODEM STATUS INTERRUPT?
0859 CA5F08              JZ      MSINT   ; TO MODEM STATUS INTRPT ROUTINE
085C C34309              JMP     EXIT    ; UNKNOWN INTERRUPT, RETURN

085F DB06        MSINT:  IN      MSR     ; GET MODEM STATUS
0861 324210              STA     MSRSAVE ; SAVE MODEM STATUS FOR DISPATCH
0864 4F                  MOV     C,A     ; SAVE IT
0865 E601                AN1     DCTS    ; HAS CTS CHANGED?
0867 C46D08              CNZ     CTSINT  ; YES GO HANDLE CTS CHANGE
086A C34309              JMP     EXIT

086D 79          CTSINT: MOV     A,C     ; GET MODEM STATUS BACK
086E E610                AN1     CTS     ; TEST CTS BIT
0870 CA7608              JZ      DISABLETX       ; OFF, DISABLE TRANSMIT
0873 C37D08              JMP     ENABLETX        ; TRY TO ENABLE TRANSMIT

                 DISABLETX:
0876 DB01                IN      IER     ; GET INTERRUPT ENABLE REGISTER
0878 E6FD                AN1     0FFH-ETBEI
087A D301                OUT     IER     ; TURN OFF TRANSMIT INTERRUPTS
087C C9                  RET

                 ENABLETX:
087D 3A4310              LDA     INTFLAG ; IS TRANSMITTER BUSY?
0880 E602                ANI     TXBUSY
0882 C8                  RZ              ; NO, RETURN
0883 DB01                IN      IER     ; GET INTERRUPT ENABLE REGISTER
0885 F602                OR1     ETBEI
0887 D301                OUT     IER     ; ENABLE TRANSMIT INTERRUPTS
0889 C9                  RET
                 *********************************************************
088A DB00        RXINT:  IN      RBR     ; READ DATA FROM SERIAL PORT
088C 2A4A10              LHLD    RNEXT   ; GO TO ROUTINE ADDRESS IN RNEXT
088F E9                  PCHL

0890 FE10        RSTART: CPI     DLE     ; IS IT A DATA LINK ESCAPE?
0893 C24309              JNZ     EXIT    ; NO
0895 219E08              LX1     H,RSTX  ; YES, NOW WAIT FOR START OF TEXT
0898 224A10              SHLD    RNEXT
089B C34309              JMP     EXIT

089E FE02        RSTX:   CPI     STX     ; IS IT START OF TEXT
08A0 C2AC08              JNZ     RSTX1   ; NO
08A3 21B508              LX1     H,RDATA ; YES, HANDLE TRANSPARENT DATA
08A6 224A10              SHLD    RNEXT
08A9 C34309              JMP     EXIT
08AC 219008      RSTX1:  LXI     H,RSTART        ; FALSE START GO BACK
08AF 224A10              SHLD    RNEXT           ; TO BEGINNING
08B2 C34309              JMP     EXIT

08B5 FE10        RDATA:  CPI     DLE     ; IS IT A DLE?
08B7 CAC308              JZ      RDATA1  ; YES
08BA CDEE08              CALL    RPUT    ; NO, PUT DATA INTO BUFFER
08BD CA0609              JZ      RESTART ; ERROR, RESET BUFFER AND RESTART
08C0 C34309              JMP     EXIT            ; FROM BEGINNING
08C3 21CC08      RDATA1: LXI     H,RCONTROL      ; RECEIVE CONTROL
08C6 224A10              SHLD    RNEXT           ; CHARACTER NEXT
08C9 C34309              JMP     EXIT

                 RCONTROL:
08CC FE10                CPI     DLE     ; IS IT A SECOND DLE?
08CE C2E008              JNZ     RCONTROL1       ; NO, CHECK FOR ETX
08D1 CDEE08              CALL    RPUT    ; YES, PUT DLE IN BUFFER
08D4 CA0609              JZ      RESTART ; ERROR, RESET BUFFER AND RESTART
08D7 21B508              LX1     H,RDATA ; GO BACK FOR MORE DATA
08DA 224A10              SHLD    RNEXT
08DD C34309              JMP     EXIT
                 RCONTROL1:
08E0 FE03                CPI     ETX     ; IS IT END OF TEXT?
```

```
08E2 C20609            JNZ     RESTART ; NO, ERROR - RESTART
08E5 211D09            LXI     H,R1CRC ; NEXT RECEIVE FIRST CRC CHAR
08E8 224A10            SHLD    RNEXT
08EB C34309            JMP     EXIT

08EE 4F       RPUT:    MOV     C,A      ; SAVE DATA IN REGISTER C
08EF 2A0610            LHLD    OTBE     ; PUT DATA INTO BUFFER
08F2 EB                XCHG
08F3 2A0810            LHLD    TBIP
                       INCTB   1
08F6+3E01              MVI     A,1
08F8+D7                RST     2
08F9 C8                RZ      ; RETURN WITH ZERO STATUS IF OVERFLOW
08FA 220810            SHLD    TBIP     ; UPDATE POINTER IF OK
08FD 71                MOV     M,C      ; MOVE DATA INTO BUFFER
08FE 211C10            LX1     H,BUFCOUNT      ; INCREMENT COUNT OF DATA
0901 34                INR     M
0902 7E                MOV     A,M
0903 FEFB              CPI     251      ; HAVE WE GOT 251 BYTES NOW?
0905 C9                RET              ; ZERO STATUS IF TOO MANY BYTES

0906 3E00     RESTART:MVI      A,0      ; SET COUNT IN BUFFER TO ZERO
0908 321C10            STA     BUFCOUNT
090B 2A0410            LHLD    CTBIE    ; SET INPUT POINTER JUST BEFORE
                       INCTB   1        ; DATA AREA
090E+3E01              MVI     A,1
0910+D7                RST     2
0911 220810            SHLD    TBIP
0914 219008            LX1     H,RSTART        ; AND RESTART RECEIVER
0917 224A10            SHLD    RNEXT
091A C34309            JMP     EXIT

091D 324710   R1CRC:   STA     RCRC1    ; SAVE FIRST CRC CHARACTER
0920 212909            LX1     H,R2CRC ; NOW GET SECOND CRC CHARACTER
0923 224A10            SHLD    RNEXT
0926 C34309            JMP     EXIT

0929 324610   R2CRC:   STA     RCRC2    ; SAVE SECOND CRC CHARACTER
092C DB04              IN      MCR      ; RESET REQUEST TO SEND
092E E6FD              AN1     0FFH-RTS
0930 D304              OUT     MCR
0932 3A4310            LDA     INTFLAG         ; INDICATE RECEIVE ROUTINE
0935 E6FE              AN1     0FFH-RXBUSY     ; IS NOT ACTIVE
0937 324310            STA     INTFLAG
093A 214309            LX1     H,EXIT   ; IGNORE ALL RECEIVE INTERRUPTS
093D 224A10            SHLD    RNEXT
0940 C34309            JMP     EXIT
0943 C1       EXIT:    POP     B
0944 D1                POP     D
0945 El                POP     H
0946 Fl                POP     PSW
0947 FB                EI
0948 C9                RET
              ***********************************************************
              ; TRANSMIT INTERRUPT ROUTINES

0949 2A4C10   TXINT:   LHLD    TNEXT    ; DISPATCH ADDRESS IN TNEXT
094C E9                PCHL

0940 3E16     TSTART:  MVI     A,SYN    ; OUTPUT A SYN CHARACTER
094F D300              OUT     THR
0951 215A09            LX1     H,TDLE1 ; NEXT SEND A DLE
0954 224C10            SHLD    TNEXT
0957 C34309            JMP     EXIT

095A 3E10     TDLE 1:  MVI     A,DLE    ; OUTPUT A DLE
095C D300              OUT     THR
095E 216709            LXI     H,TSTX   ; NEXT OUTPUT START OF TEXT
0961 224C10            SHLD    TNEXT
0964 C34309            JMP     EXIT

0967 3E02     TSTX:    MVI     A,STX    ; OUTPUT START OF TEXT
0969 D300              OUT     THR
096B 217409            LX1     H,TDATA ; NEXT FUNCTION HANDLES TEXT DATA
096E 224C10            SHLD    TNEXT
0971 C34309            JMP     EXIT

0974 211D10   TDATA:   LXI     H,OUTCOUNT      ; MORE DATA IN BUFFER?
0977 7E                MOV     A,M
0978 B7                ORA     A
0979 CA9709            JZ      TDATA1   ; NO, BUFFER EMPTY
097C 35                DCR     M
097D 2A1A10            LHLD    LBOP
                       INCLB   1
0980+3E01              MVI     A,1
0982+DF                RST     3
0983 221A10            SHLD    LBOP     ; LBOP = LBOP+1
0986 7E                MOV     A,M
0987 D300              OUT     THR      ; OUTPUT DATA AT LBOP
0989 FE10              CPI     DLE      ; IS IT SAME AS DLE?
098B C24309            JNZ     EXIT     ; NO
098E 21A409            LXI     H,TDLE   ; TRANSMIT ANOTHER DLE
0991 224C10            SHLD    TNEXT    ; TO MAKE TRANSPARENT
0994 C34309            JMP     EXIT
0997 3E10     TDATA1:  MVI     A,DLE    ; OUTPUT A DATA LINK ESCAPE
0999 D300              OUT     THR
099B 21B109            LXI     H,TETX   ; NEXT SEND END OF TEXT
099E 224C10            SHLD    TNEXT
09A1 C34309            JMP     EXIT

09A4 3E10     TDLE:    MVI     A,DLE    ; SEND DATA LINK ESCAPE
09A6 D300              OUT     THR
09A8 217409            LXI     H,TDATA ; AND GO BACK TO TRANSPARENT MODE
09AB 224C10            SHLD    TNEXT
09AE C34309            JMP     EXIT

09B1 3E03     TETX:    MVI     A,ETX    ; SEND END OF TEXT
09B3 D300              OUT     THR
09B5 21BE09            LX1     H,T1CRC ; NEXT SEND FIRST CRC CHARACTER
09B8 224C10            SHLD    TNEXT
09BB C34309            JMP     EXIT

09BE 3A4910   T1CRC:   LDA     TCRC1    ; SEND FIRST CRC CHARACTER
09C1 D300              OUT     THR
09C3 21CC09            LXI     H,T2CRC ; NEXT SEND SECOND CRC CHARACTER
09C6 224C10            SHLD    TNEXT
09C9 C34309            JMP     EXIT

09CC 3A4810   T2CRC:   LDA     TCRC2    ; SEND SECOND CRC CHARACTER
09CF D300              OUT     THR
09D1 21DA09            LX1     H,TPAD   ; SEND TRAILING PAD CHARACTER
09D4 224C10            SHLD    TNEXT
09D7 C34309            JMP     EXIT

09DA 3EFF     TPAD:    MVI     A,PAD    ; SEND TRAILING PAD AFTER CRC
09DC D300              OUT     THR
09DE 3A4310            LDA     INTFLAG ; MARK TRANSMIT NOT BUSY
09E1 E6FD              AN1     0FFH-TXBUSY
09E3 324310            STA     INTFLAG
09E6 CD7608            CALL    DISABLETX
09E9 C34309            JMP     EXIT
              ***********************************************************
              ; CRC CALCULATION ROUTINE
              ; INCLUDES BYTE IN ACCUMULATOR IN CRC CALCULATION
09EC F5       CALCCRC:PUSH     PSW
09ED 0608              MVI     B,8
09EF 4F                MOV     C,A
09F0 2A4410            LHLD    CRC
09F3 =        CALCCRC1:EQU     $
09F3 79                MOV     A,C
09F4 07                RLC
09F5 4F                MOV     C,A
09F6 7D                MOV     A,L
09F7 17                RAL
```

```
09F8 6F              MOV     L,A
09F9 7C              MOV     A,H
09FA 17              RAL
09FB 67              MOV     H,A
09FC D2070A          JNC     CALCCRC2
09FF 7C              MOV     A,H
0A00 EE80            XRI     80H
0A02 67              MOV     H,A
0A03 7D              MOV     A,L
0A04 EE05            XRI     05H
0A06 6F              MOV     L,A
0A07 =     CALCCRC2:EQU     $
0A07 05              DCR     B
0A08 C2F309          JNZ     CALCCRC1
0A0B 224410          SHLD    CRC
0A0E F1              POP     PSW
0A0F C9              RET
           ***************************************************
           ; RECEIVE SIDE DISPATCH ROUTINES

0A10 2A4E10  DISPRX: LHLD    RDISP   ; GO TO RECEIVE DISPATCH ROUTINE
0A13 E9              PCHL

0A14 2A0610  WAITTB: LHLD    OTBE    ; TERMINAL BUFFER CUSHION FREE?
0A17 EB              XCHG
0A18 2A0410          LHLD    CTBIE
                     COMPARE         ; COMPARE DE TO HL
0A1B+EF              RST     5
0A1C CA280A          JZ      WAITTB1 ; SAME, BUFFER AVAILABLE
                     INCTB   CUSHION ; IS CUSHION FREE?
0A1F+3EFF            MVI     A,CUSHION
0A21+D7              RST     2
0A22 DA120B          JC      DISPTX  ; TO TRANSMIT ROUTINE DISPATCHER
0A25 2A0410          LHLD    CTBIE   ; POINT TBIP JUST AHEAD OF DATA
           WAITTB1:INCTB     1
0A28+3E01            MVI     A,1
0A2A+D7              RST     2
0A2B 220810          SHLD    TBIP
0A2E 3E00            MVI     A,0     ; ZERO COUNT FOR RECEIVE ROUTINE
0A30 321210          STA     BUFCOUNT
0A33 F3              DI
0A34 3A4310          LDA     INTFLAG ; RECEIVE ROUTINE IS ACTIVE
0A37 F601            OR1     RXBUSY
0A39 324310          STA     INTFLAG
0A3C FB              EI
0A3D 219008          LX1     H,RSTART        ; START RECEIVING
0A40 224A10          SHLD    RNEXT
0A43 DB04            IN      MCR     ; SET RTS SO OTHER END WILL SEND
0A45 F602            OR1     RTS
0A47 D304            OUT     MCR
0A49 21500A          LX1     H,WAITRX        ; WAIT FOR BLOCK
0A4C 224E10          SHLD    RDISP
0A4F C9              RET

0A50 3A4310  WAITRX: LDA     INTFLAG ; IS RECEIVER STILL BUSY?
0A53 E601            AN1     RXBUSY
0A55 C2120B          JNZ     DISPTX  ; YES, GO TO TRANSMIT DISPATCHER
0A58 3A5210          LDA     DFLAG   ; GET DISPATCHER FLAG
0A5B E601            AN1     CRCTX   ; IS CRC ROUTINE BUSY?
0A5D C2120B          JNZ     DISPTX  ; YES, GO TO TRANSMIT DISPATCHER
0A60 211C10          LX1     H,BUFCOUNT      ; COUNT OF BYTES RECEIVED
0A63 7E              MOV     A,M
0A64 2A0410          LHLD    CTBIE   ; POINT TO CURRENT INPUT ENTRY
0A67 77              MOV     M,A     ; PUT COUNT IN BUFFER HEADER
                     INCTB   1       ; POINT JUST BEFORE DATA AREA
0A68+3E01            MVI     A,1
0A6A+D7              RST     2
0A6B 220810          SHLD    TBIP
0A6E 210000          LX1     H,0     ; INITIALIZE CRC VALUE
0A71 224410          SHLD    CRC
0A74 217B0A          LX1     H,RXCRC ; NEXT CALCULATE CRC
0A77 224310          SHLD    RDISP
```

```
0A7A C9              RET

0A7B 211C10  RXCRC:  LXI     H,BUFCOUNT      ; MORE DATA TO INCLUDE?
0A7E 7E              MOV     A,M
0A7F B7              ORA     A
0A80 CA970A          JZ      RXCRC1  ; NO, GO TO INCLUDE CONTROL CHARS
0A83 35              DCR     M       ; DECREMENT COUNT
0A84 2A0810          LHLD    TBIP    ; UPDATE POINTER TO NEXT POSITION
                     INCTB   1
0A87+3E01            MVI     A,1
0A89+D7              RST     2
0A8A 220810          SHLD    TBIP
0A8D 7E              MOV     A,M     ; GET DATA BYTE IN A
0A8E CDEC09          CALL    CALCCRC ; INCLUDE IT IN CRC CALCULATION
0A91 FE10            CPI     DLE     ; WAS IT DATA LIKE A DLE?
0A93 CCEC09          CZ      CALCCRC ; DO ANOTHER FOR TRANSPARENCY
0A96 C9              RET             ; RETURN TO LIP
0A97 3E10    RXCRC1: MVI     A,DLE   ; INCLUDE DLE AND ETX IN CRC
0A99 CDEC09          CALL    CALCCRC
0A9C 3E03            MVI     A,ETX
0A9E CDEC09          CALL    CALCCRC
0AA1 21A80A          LXI     H,CHECKCRC      ; NEXT CHECK THE CRC
0AA4 224E10          SHLD    RDISP   ; GO INCLUDE CRC CHARS RECEIVED
0AA7 C9              RET

0AA8 3A4710  CHECKCRC:LDA    RCRC1   ; INCLUDE RECEIVED CRC CHARACTERS
0AAB CDEC09          CALL    CALCCRC
0AAE 3A4610          LDA     RCRC2
0AF31 CDEC09         CALL    CALCCRC
0AB4 21BB0A          LXI     H,CHKFIN        ; NEXT CHECK IF CRC IS GOOD
0AB7 224310          SHLD    RDISP
0ABA C9              RET

0ABB 2A4410  CHKFIN: LHLD    CRC     ; GET CALCULATED CRC
0ABE 7D              MOV     A,L     ; IS IT ZERO?
0ABF B4              ORA     H
0AC0 C2D00A          JNZ     CHKFIN1 ; NO, GO RESTART RECEIVE OPERATION
0AC3 DB04            IN      MCR     ; YES, GOOD CRC, FLIP DTR
0AC5 EE01            XRI     DTR
0AC7 D304            OUT     MCR
0AC9 21D70A          LXI     H,RPROC ; PROCESS CHECKED BLOCK
0ACC 224310          SHLD    RDISP
0ACF C9              RET
0AD0 21140A  CHKFIN1:LXI     H,WAITTB        ; BAD CRC, TRY AGAIN
0AD3 224E10          SHLD    RDISP
0AD6 C9              RET

             ; THIS ROUTINE SHOULD PROCESS THE BUFFER PREFIX
             ; TEMPORARILY IT ONLY PASSES THE BUFFER TO THE LIP
             ; AND HANDLES CONNECT/DISCONNECT
0AD7 2A0410  RPROC:  LHLD    CTBIE   ; IS COUNT 7 OR MORE?
0ADA 7E              MOV     A,M
0ADB FE07            CPI     7
0ADD DA020B          JC      RPROC2  ; NO, PASS TO LIP
                     INCTB   8       ; POINT TO SEE IF CONNECT OR
0AE0+3E08            MVI     A,8
0AE2+D7              RST     2
0AE3 7E              MOV     A,M     ; DISCONNECT
0AE4 FE01            CPI     'A'-40H ; IS IT CONNECT?
0AE6 C2F30A          JNZ     RPROC1  ; NO, GO TO TEST FOR DISCONNECT
0AE9 3E00            MVI     A,0     ; 0 FOR CONNECT
0AEB F7              RST     6       ; COMMUNICATE REQUEST TO LIP
0AEC 21140A          LXI     H,WAITTB        ; DON'T PASS THIS ENTRY
0AEF 224E10          SHLD    RDISP
0AF2 C9              RET
0AF3 FE02    RPROC1: CPI     'B'-40H ; IS IT DISCONNECT?
0AF5 C2020B          JNZ     RPROC2  ; NO, PASS TO LIP
0AF8 3E01            MVI     A,1     ; YES, 1 FOR DISCONNECT
0AFA F7              RST     6       ; COMMUNICATE REQUEST TO LIP
0AFB 21140A          LXI     H,WAITTB        ; DON'T PASS THIS ENTRY
0AFE 224310          SHLD    RDISP
0B01 C9              RET
```

```
0B02 2A0810      RPROC2: LHLD    TBIP    ; UPDATE CURRENT INPUT ENTRY
                         INCTB   1
0B05+3E01                MVI     A,1
0B07+D7                  RST     2
0B08 220410              SHLD    CTBIE
0B0B 21140A              LX1     H,WAITTB        ; NOW GO GET ANOTHER ONE
0B0E 224E10              SHLD    RDISP
0B11 C9                  RET
                 ********************************************************
                 ; TRANSMIT SIDE DISPATCH ROUTINES

0B12 2A5010      DISPTX: LHLD    TDISP   ; GO TO TRANSMIT DISPATCH ROUTINE
0B15 E9                  PCHL

                 ; THIS ROUTINE SHOULD PROCESS THE BUFFER PREFIX BUT
                 ; TEMPORARILY IT ONLY PASSES THE BUFFER TO THE HOST
                 WAITLIP:NEXTIN  ; IS THERE A BUFFER ENTRY FROM THE LIP?
0B16+E7                  RST     4
0B17 C8                  RZ              ; NO, RETURN
0B18 3A5210              LDA     DFLAG   ; INDICATE TX SIDE USING CRC
0B1B F601                OR1     CRCTX   ; ROUTINES
0B1D 325210              STA     DFLAG
0B20 7E                  MOV     A,M     ; GET DATA LENGTH FROM HEADER
0B21 321D10              STA     OUTCOUNT ; FOR CRC CALCULATION ROUTINE
                         INCLB   3       ; POINT JUST BEFORE DATA AREA
0B24+3E03                MVI     A,3
0B26+DF                  RST     3
0B27 221A10              SHLD    LBOP    ; FOR CRC CALCULATION ROUTINE
0B2A 210000              LX1     H,0     ; INITIALIZE CRC VALUE
0B2D 224410              SHLD    CRC
0B30 21370B              LX1     H,TXCRC ; NEXT START CRC CALCULATION
0B33 225010              SHLD    TDISP
0B36 C9                  RET

0B37 211D10      TXCRC:  LXI     H,OUTCOUNT    ; ANY MORE DATA TO INCLUDE?
0B3A 7E                  MOV     A,M
0B3B B7                  ORA     A
0B3C CA530B              JZ      TXC RC 1  ; NO, GO TO INCLUDE CONTROL CHARS
0B3F 35                  DCR     M         ; DECREMENT COUNT
0B40 2A1A10              LHLD    LBOP      ; UPDATE POINTER TO NEXT POSITION
                         INCLB   1
0B43+3E01                MVI     A,1
0B45+DF                  RST     3
0B46 221A10              SHLD    LBOP
0B49 7E                  MOV     A,M       ; GET DATA BYTE IN A
0B4A CDEC09              CALL    CALCCRC ; INCLUDE IT IN CRC CALCULATION
0B4D FE10                CPI     DLE     ; WAS IT DATA LIKE A DLE?
0B4F CCEC09              CZ      CALCCRC ; DO ANOTHER FOR TRANSPARENCY
0B52 C9                  RET             ; RETURN TO LIP
0B53 3E10        TXCRC1: MVI     A,DLE   ; INCLUDE DLE AND ETX IN CRC
0B55 CDEC09              CALL    C ALC C RC
0B58 3E03                MVI     A,ETX
0B5A CDEC09              CALL    CALCCRC
0B5D 21640B              LXI     H,CRCFIN ; NEXT TO FINISH CRC FOR SENDING
0B60 225010              SHLD    TDISP
0B63 C9                  RET

0B64 3E00        CRCFIN: MVI     A,0       ; FINISH OFF CRC CALCULATION FOR
0B66 CDEC09              CALL    CALCCRC   ; TRANSMISSION
0B69 CDEC09              CALL    CALCCRC
0B6C 2A4410              LHLD    CRC       ; SAVE CALCULATION FOR TRANSMIT
0B6F 224810              SHLD    TCRC2
0B72 21790B              LX1     H,STARTTX        ; NEXT, START TRANSMITTING
0B75 225010              SHLD    TDISP            ; THE BLOCK
0B78 C9                  RET

0B79 3A4210      STARTTX:LDA     MSRSAVE  ; SAVE CURRENT DSR LEVEL
0B7C E620                AN1     DSR
0B7E 67                  MOV     H,A
0B7F 3A5210              LDA     DFLAG
0B82 E6DE                AN1     0FFH-CRCTX-DSR   ; INDICATE CRC ROUTINE
0B84 B4                  ORA     H                ; NOT IN USE AND SAVE
0B85 325210              STA     DFLAG            ; DSR IN DFLAG
0B88 214D09              LX1     H,TSTART         ; SET UP INITIAL XMIT
0B8B 224C10              SHLD    TNEXT            , INTERRUPT ROUTINE
0B8E 2A1610              LHLD    OLBE      ; POINT TO DATA TO TRANSMIT
0B91 7E                  MOV     A,M       ; GET COUNT
0B92 321D10              STA     OUTCOUNT
                         INCLB   3
0B95+3E03                MVI     A,3
0B97+DF                  RST     3
0B98 221A10              SHLD    LBOP
0B9B F3                  DI              ; DISABLE INTERRUPTS
0B9C 3A4310              LDA     INTFLAG ; INDICATE TRANSMIT BUSY
0B9F F602                OR1     TXBUSY
0BA1 324310              STA     INTFLAG
0BA4 3A4210              LDA     MSRSAVE ; IS CTS UP?
0BA7 E610                AN1     CTS
0BA9 CAB20B              JZ      STARTTX1 ; DON'T ENABLE TRANSMIT INTRPT
0BAC DB01                IN      IER     ; YES, ENABLE TRANSMIT INTERRUPTS
0BAE F602                OR1     ETBEI
0BB0 D301                OUT     IER
0BB2 FB          STARTTX1:EI             ; ENABLE INTERRUPTS
0BB3 21BA0B              LX1     H,WAITTX ; WAIT FOR TRANSMIT TO FINISH
0BB6 225010              SHLD    TDISP
0BB9 C9                  RET

0BBA 3A4310      WAITTX: LDA     INTFLAG ; TRANSMITTER INTERRUPTS ENABLED?
0BBD E602                ANI     TXBUSY
0BBF C0                  RNZ             ; YES, RETURN
0BC0 211027              LX1     H,ACKTO ; NO, SET UP FOR TIMEOUT
0BC3 224010              SHLD    WAIT    ; INITIALIZE ACKNOWLEDGE TIMEOUT
0BC6 21CD0B              LX1     H,WAITACK      ; NEXT WAIT FOR ACKNOWLEDGE
0BC9 225010              SHLD    TDISP
0BCC C9                  RET

                 WAITACK:
0BCD 215210              LX1     H,DFLAG ; IS DSR SAME AS BEFORE?
0BD0 3A4210              LDA     MSRSAVE
0BD3 AE                  XRA     M
0BD4 E620                ANI     DSR
0BD6 C2EA0B              JNZ     WAITACK1       ; NO, BLOCK ACKNOWLEDGED
0BD9 2A4010              LHLD    WAIT    ; YES, DECREMENT TIMEOUT COUNT
0BDC 2B                  DCX     H
0BDD 224010              SHLD    WAIT
0BE0 7C                  MOV     A,H     ; IS TIME OVER?
0BE1 B5                  ORA     L
0BE2 C0                  RNZ             ; NO, RETURN
0BE3 217908              LX1     H,STARTTX ; YES, TIMED OUT, SO SEND AGAIN
0BE6 225010              SHLD    TDISP
0BE9 C9                  RET
                 WAITACK1:
0BEA 21160B              LXI     H,WAITLIP  ; GOOD ACK, GET ANOTHER BUFFER
0BED 225010              SHLD    TDISP      ; FROM LIP
0BF0 C9                  RET

0BF1                     END
```

LINK LEVEL ADDRESS MECHANISMS IN AMATEUR RADIO PROTOCOLS

by

H. S. Magnuski, KA6M
311 Stanford Avenue
Menlo Park, CA 94025

In October, 1982, agreement was reached on a new Link Level protocol for amateur packet radio networks. One of the unique aspects of the protocol is the set of address fields used at the beginning of each frame. This paper reviews the types of addressing used prior to the adoption of the new standard, and explains in detail how the new address mechanism works.

On October 9th, 1982, representatives of the active packet radio user groups throughout the U.S. agreed to establish a new standard for Link Level connections in the packet radio service. This agreement unified the development activities of several diverse groups, and provides for a point-to-point protocol which can be used both in terrestrial and satellite networking, This paper will review how addressing was done prior to the adoption of the new standard, and will then describe, in some detail, how the new address mechanism works, and what advantages and disadvantages are gained by adopting it.

1.0 Background of HDLC Numerical Addressing

The first byte of an HDLC frame following the opening flag is always an address byte. This byte permits a maximum of 256 stations in a network to be addressed, and sometimes less, as byte 00 hex is often reserved for special purposes, and byte FF hex is usually a broadcast (all-stations-addressed) address. Since 254 stations is a reasonable number for active stations on a single frequency metropolitan area net (metronet), the initial implementor of HDLC-oriented transmission, Doug Lockhart) VE7APU, decided that no more complicated addressing scheme was required and he programmed his Terminal 'Node Controller board to deal with a single byte address. Doug's scheme initially utilized dynamic address assignment) and works as follows:

When a station initially came on the air, it would send a special sign-on packet to a central control station, which would assign the next available numerical address to the new user. If the user logged off or timed out, the address would be put back into the free pool and handed to the next station to sign in. Once assigned an address, the station would use that address in all outgoing packets, and the central control station would know by that address who originated each packet in the metronet. All traffic flowed through the central control station, so there was no need for one remote to address another,

Several groups in Canada and the U.S. desired to use Doug's code without a central station, and so the following form of static addressing was adopted:

Each user in a specific geographical area was assigned a numeric address, and that address was hard-wired into the outgoing address field of each transmitted frame. After a connection was made, the receiving station would lock onto the transmitting station's address, and all other packets on the channel would be discarded.

The author implemented a variation of this idea that would allow a simplex repeater to be used as part of a metronet. Some range of the address space, say 80 to 9F hex would be dedicated as repeater input addresses, and in repeating the packet the repeater would transform the address to the range A0 to BF. Thus a station would use address 01 for point-to-point work, and would use addresses 81 and A1 for repeater uplink and downlink, respectively.

The dynamic and static addressing; schemes described above served the early packet experimenters well, but it soon became clear that there were significant problems with each of the methods.

The dynamic addressing scheme failed if the central control station failed, and many users did not want to be dependent on a single central repeater, The scheme did not allow for multiple repeaters serving a given area, and failed if there was any overlap in central or remote station RF domains,

The static assignment method was just as bad, requiring an individual or club to prescribe addresses, and limiting the number of users to 62 if a repeater was involved. Visitors to an area might conflict with already assigned addresses, and using statically assigned addresses on a global satellite channel would be impossible,,

Since neither of these methods would be workable over the long term, several proposals were put forward to correct the problem.

2.0 HDLC Extended Addressing

It became clear that if one could substitute a callsign for the hardwired numeric code, some of the addressing problems mentioned above would go away. The radio callsign is a unique identifier, and could be used to tag each packet being transmitted. So Terry Fox and others at AMRAD in Washington, D.C., proposed to utilize the extended addressing feature in HDLC to incorporate the transmitting callsign.

The HDLC standard permits an arbitrary extension of the address field through utilization of the least significant bit of the address byte as an extension flag. Basically, if the least significant bit (lsb) is zero, then the next byte forms part of the address field too, and one continues to extend the address field until a byte is found where the lsb is set to one.

This idea would be fairly easy to implement, and did not require much overhead in each packet. It also did not violate any standards related to the HDLC spec. Unfortunately, there were deficiencies in the method, particularly with regard to point-to-point addressing in a network, and flexible use of multiple repeaters in one RF domain. With just one address, a station could not target the receiver of an outgoing packet. Also, no provisions were made for dealing with multiple repeaters on the same frequency, and it became clear that this approach would have to be revised.

3.0 The New Link Level Standard

When the packet radio groups got together in October there was clearly an urgent need to agree upon a uniform link level standard, and to adopt one which would be sufficiently flexible to accomodate a variety needs and environments.

Two proposals were put forward at the meeting, and it turned out that there was more in common in the proposed solutions than there were differences. The first proposal was by the AMRAD and New Jersey packeteers, and it was a result of extensive discussions on how to implement an X.25 type of service suitable for the radio community. This proposal, called AX.25, called for a destination and source callsign in an extended address field. The other proposal was from the author, and it also specified use of two or optionally three callsigns in the packet, but positioned them immediately after the control field. Each party was willing to compromise, the callsigns were kept in the extended address field, an optional repeater address field was added to the AX.25 spec, definitions for bits in the sub-station and protocol ID fields were rearranged, and thus emerged the new standard protocol.

4.0 Components of the Address Field

As stated before, the standard protocol relies on the HDLC extended addressing bit to mark the end of the address field. The address field is composed of two or three seven-byte callsign address fields constructed in the following manner:

CHAR1 CHAR2 CHAR3 CHAR4 CHAR5 CHAR6 SSTD

where each call is left justified in the field, padded with blanks, uppercase, and each CHAR is shifted left so that the lsb is zero, except for the last byte of the address fields, where the lsb is set to one, indicating the end end of HDLC extended addressing. The first callsign field is for the destination or receiving station, the second callsign field is for the source or transmitting station, and the optional third field is for a repeater.

The SSID byte is of the form:

R 11 SSSS X

where R is the "repeated" bit, set to one only in the optional repeater address field when the packet has been repeated. The '11' field is reserved and set to ones. It may be used for control purposes if required by a local area net. The SSSS field is the sub-station code, normally all zeros except for situations where a person has more than one station on the air. The X is normally zero, unless this field is the last of the callsigns, in which case it is set to one, indicating the end of the variable length address field.

5.0 Uses of the Address Fields

The simplest case is a point-to-point connection between two stations. Station A puts CALL B in the destination field, and CALL A in the source field. Similarly, Station B puts CALL A in the destination field and CALL B in the source field. When A receives a packet it first verifies that the frame check sequence (FCS) is correct. Next, it scans the address field to determine if the address field length is either 14 or 21 bytes. Assuming that the field is exactly 14 bytes long, it checks for CALL A in the destination field and for CALL B in the source field, and discards the packet if there is not a correct match.

If a repeater is being used, the optional third address field will be utilized, and the

following sequence of actions will take place:

The transmitting station extends the address field to 21 bytes and places the callsign of the desired repeater into the third field. The I've-been-repeated bit (R bit) must be set to zero. The repeater is constantly monitoring the channel for packets and discards all packets which do not have a correct FCS or do not have exactly 21 bytes in the address field. If the third address field is present and if the call exactly matches the repeater's call and SSID, and if the R bit is zero, then the repeater sets the R bit to one and retransmits the packet. The receiving station determines that the address field is exactly 21 bytes and checks that the R bit is set to one. If the R bit is zero it has received an uplink packet and should discard it. The receiving station then checks for proper source and destination callsigns, and accepts the packet on a proper match.

The receiving station knows how to reply to an incoming transmission because an examination of the address field will inform it whether or not a repeater was used, and if one was used, its callsign. This information is used to construct the reverse path to the transmitting station.

6.0 Limitations

This protocol is intended only for point-to-point connections directly or through a single repeater. The situation where multiple repeats are required is not covered and is the task of higher level protocols.

The protocol also does not specify when the reverse path should be constructed. An implementor may decide to establish the reverse path only on information contained in the initial connect (SABM) packet. If so, and if the connector changes paths during a session, the two stations might be using different forward and reverse paths to talk to each other.

The call signs add overhead to each packet (20 bytes at most), and these extra bytes will obviously slow down throughput. In doing link calculations, however, one finds that radio and modem turnaround delays are probably the most significant factors affecting overall efficiency, and that the extra overhead of the callsigns are justified in terms of the elimination of the requirement for a central control station, or elimination of any kind of connection-state information in the repeater.

7.0 Summary

The address mechanisms in this new protocol are fairly simple to implement and provide for a point-to-point connection in local area metronets, in backbone terrestrial networks, and in satellite channels. The addition of a single repeater option makes the protocol useful for communications in a limited geographical area. The protocol will serve as a building block for higher level connection-oriented protocols such as AX.25, and can be easily used for connectionless, datagram-oriented protocols such as IP/TCP.

REAL-TIME LOW LEVEL SOFTWARE ON THE TAPR TNC

Margaret Morrison, KV7D
4301 E. Holmes Street
Tucson, Arizona 85711
602-325-4775

Introduction

This paper describes the low-level assembly language routines (LLR) of software released with the Tucson Amateur Packet Radio (TAPR) Beta Test terminal node controllers (TNCs). The primary functions performed by these routines are initialization of peripheral devices and data in RAM, maintaining input and output (I/O) buffers, servicing interrupts from peripheral devices, handling nonvolatile RAM data storage and retrieval, and calibration and checkout routines. Entry points are provided which are appropriate to the subroutine calling sequence of the Pascal compiler used for the high-level routines (HLR). In addition, a low-level debug program provides capability for direct access to peripherals, inspection of RAM and ROM locations, and execution of temporary code in RAM. The present LLR code occupies about six kilobytes of ROM.

Initialization

On receipt of a RESTART interrupt by the processor, control passes to the low-level initialization routine. This section first sets up the hardware stack, and then reads the on-board DIP switches which direct the initialization of user-settable parameters. These parameters, which can be set to their default values stored in ROM, or read from the 64 x 4 nonvolatile RAM, are decoded and expanded. The program initializes buffer pointers, counters, timers, and status flags, as well as non-permanent user-settable parameters. The peripheral chips are initialized and checked to verify that they can be commanded. If the UART can not be commanded, the program is aborted and another reset is attempted. Failure of other peripherals results in diagnostic messages. The VIA timer functions are configured so that Timer 1 acts as a pulse generator for the HDLC controller and Timer 2 generates interrupts for software timing functions. The initialization section terminates by enabling interrupts and typing a sign-on message, and passes control to the HLR.

Buffer Management

One of the primary tasks of the LLR is maintaining the I/O buffers. At any time there are four active I/O buffers, input and output buffers for terminal and radio data. An echo buffer, which is configured identically with the terminal output buffer may also be present. Data received from peripherals (terminal and radio interface) are placed into input buffers which are read upon calls from HLR. Data placed into output buffers by calls from HLR are passed to the peripherals under interrupt control. Each buffer has an insertion pointer, which is updated as data are added to the buffer so that it points to the next available cell, and a removal pointer which points to the next cell to be read. All buffers are "circular", meaning that each time a pointer is advanced it is compared with the top of the buffer space, and moved if necessary to the bottom of the buffer space. A buffer is "empty" when the insertion and removal pointers are the same, and "full" when the insertion pointer points to the next cell below the removal pointer. The incoming and outgoing packet buffers contain, as the first bytes of each packet, the byte count of the packet.

In addition to the two basic pointers, input buffers have other markers to facilitate input editing, and output buffers have space-available counters for use by routines writing to these buffers.

A pointer to the beginning of an incoming packet serves two purposes. First, it allows an incoming packet to be purged in case of a reception error (invalid frame-check sequence or insufficient buffer space). Second, it facilitates storing the byte count of the packet upon successful reception.

The terminal input buffer management is rather complex, and operates in three different modes. In "command mode," character and line editing functions are in effect,, and a pointer to the beginning of the current line insures that deletion past this point will not take place in case of rapid terminal input or slow input processing. In "conversation mode," an additional pointer marks the beginning of the current packet (actually the data portion of the packet), and an additional editing feature allows the user to cancel the current packet. Packets remain in the input buffer until they have been acknowledged, at which time the buffer is updated by a call from HLR. A completed packet is signalled by the receipt of a packet-terminating character, and this character is placed in the buffer as a marker for the end of the packet. In order to prevent commands from interfering with partially typed packets, separate input buffers are maintained, and the pointers for the active buffers are swapped when the mode is changed. This has the effect of moving time-consuming decisions from the interrupt dependent program to subroutines called from HLR.

The third input mode is "transparent mode," in which all characters received from the terminal are transmitted. Packets are terminated on the basis of number of characters or occurrence of a

timeout. Characters between the removal pointer and the packet marker are divided into maximum-length packets, with the remainder going into a short packet. In order to avoid ambiguity as to the composition of a packet, the packet marker is not updated until all outstanding packets have been acknowledged and cleared from the input buffer.

Interrupt Service

Although the 6809 microprocessor supports two levels of hardware interrupt, a fast interrupt (FIRQ) and a normal interrupt (IRQ), only the IRQ is implemented on the TAPR TNC. All peripheral IRQ lines are wire-ORed together into the IRQ input to the processor. Interrupt outputs from each peripheral can be disabled independently without affecting the processor's response to IRQ. Upon receipt of IRQ, the processor transfers execution to a routine which examines each device in turn and passes control to a routine specified in a dispatch table* For maximum flexibility, the interrupt dispatch table is stored in RAM. The addresses in the table are initialized during the startup procedure, and are changed as the TNC operates in different modes.

The peripheral devices are assigned priorities according to the order in which the dispatch routine checks their status. The priority of service is

1. UART (terminal) input
2. UART output
3. Timer interrupt
4. All HDLC (radio interface) interrupts

Interrupts from the parallel I/O port are not enabled in the initial software release; when they they will be assigned lowest priority. The HDLC is currently assigned lowest priority because the interrupt service is quite complex and the chip will generate spurious interrupts at a high rate under noisy conditions. The UART was placed ahead of the timer, since the 6522 timer mode used provides a mechanism for compensating for delayed interrupt servicing. The UART output ought properly to be assigned a lower priority, but it was placed after the input service for convenience, since input and output status are given in the same register. Parallel port I/O must be assigned the lowest priority to insure that it does not interfere with other interrupt service.

As a diagnostic tool, the interrupt dispatch routine writes signals reflecting interrupt conditions to the parallel port, which is configured as 16 output bits for the initial software release.

Terminal I/O

The terminal input interrupt service normally consists of two parts: an initial routine during which interrupts remain disabled, and a subsequent unprotected routine. The protected routine reads a character from the UART data register and places it in a temporary holding buffer, after which interrupts are enabled. This is because the input editing and echoing function is enabled upon receipt of each character, and this can be a complex procedure in some cases. Depending on the input mode, the character may be tested against an assortment of special characters, and this may result in flow control action or change of input mode. Characters which terminate packets or command lines require that pointers be updated and flags set for other routines. Input editing characters cause immediate update of the input buffer. Finally, an echo routine is called, which places characters in an echo output buffer. Input characters are not echoed directly, since adequate terminal support sometimes requires that more than one character be echoed for each character input. In particular, automatic line feed after carriage return is a common requirement, and some hard-copy devices require many null characters following a carriage return. If the input buffer becomes nearly full, a request may be made to the output routine to transmit an XOFF character. Alternatively, a routine to produce the appropriate hardware flow control signals to the RS-232 interface may be called.

Special input service routines are used when the program operates in "transparent mode." Since there are no special characters, the input character is simply placed in the buffer. A timer may be started, and a flag will be set if a maximum packet count has been reached.

The UART output interrupt is disabled whenever there is no data to be sent, and any routine which requests output enables the interrupt. Any special treatment of characters, such as conversion to upper case or adding line feeds or nulls, is done by the routine which fills the output buffer, which also maintains a screen-width counter and inserts extra carriage returns as necessary. The output service routine checks for tasks to be performed in the following priority and performs the first task found.

1. Request to transmit XOFF character
2. Request to transmit XON character
3. Transmit characters in echo buffer
4. Transmit characters in output buffer

If all tasks are exhausted, the routine disables the output interrupt. This is not necessary, but as long as this interrupt is enabled, the UART generates an interrupt at regular intervals.

Timer Interrupts

The basic function of the timer interrupt is to update the software clocks. The interrupt is set to occur at 10 ms intervals, which provides good resolution for timing associated with the radio interface. For longer times, a "slow" clock is updated at one-second intervals. An additional. 10 ms clock is used as a pseudo-random number generator for determining packet retry times.

In addition to the 'basic clock function, several tasks are performed under timer interrupt control. If a CW ID is in progress, the Morse Code routine is invoked every 60 ms to toggle the tone on or off as necessary. Otherwise, a variety of tasks to be performed after time lapses are checked . Packet transmissions are begun following an appropriate interval following detection of a carrier drop, and the ID is sent at regular intervals. If a CW ID is commanded manually, it is begun in this routine. A special set of routines is entered when the program runs in "transparent

mode," and the appropriate routine is selected by reference to a status table. These routines mark packets for sending upon timeout of clocks which are started on character input, and set up guard times for the escape sequence for exiting this mode.

HDLC Interrupts

The WD-1933 HDLC controller generates interrupts for seven different conditions. Since any combination of interrupt conditions can be present, and most conditions are cleared by reading the status of the chip, all possibilities must be considered at every interrupt. The interrupt conditions are requests for service of input or output registers (DRQI or DRQO), transmission or receipt of end of message, with or without errors (XEOM or REOM), and change of state of carrier detect (DSC).

Transmit interrupts, DRQO, XEOM-ok, and XEOM-err, are handled according to a state table which indicates the progress of the packet in transmission. The appropriate action, as determined by the state table, is taken if any of these interrupts is detected, without reference to which condition was indicated. The only transmit error possible is under-run, or failure to service a DRQO. This condition should never occur in normal operation. Following complete transmission of a packet and while final flags are being sent, the routine checks for further packets to be transmitted before the transmitter is unkeyed. The CW ID may also be started from this routine.

Receive interrupts, DRQI, REOM-ok, and REOM-err, are handled according to the condition indicated by the chip status. In the event that more than one condition is present, a DRQI is assumed to precede an REOM, and the input register is read before closing the packet. If both REOM-ok and REOM-err are present, the error condition is disregarded. Causes of the error condition are aborted frame, invalid frame-check sequence, and over-run (failure to service DRQI). The cause of the error is not investigated and any partially received frame is cancelled. Another possible source of error is insufficient room in the input buffer for the incoming packet. If the input buffer becomes full, a flag is set and the routine continues to read incoming characters as they appear, but an REOM-ok is treated as if it were an REOM-err.

The DSC interrupt does not affect either transmit or receive operation. The carrier detect input of the HDLC controller is the demodulator lock-detect and is used to recognize a busy channel. The service routine maintains software flags which reflect the carrier detect state, and if the carrier is found to have dropped, a timer is started which indicates when the next transmission may be started.

Nonvolatile RAM Interface

A number of user-settable parameters are stored in a semi-permanent state in the Xicor NOVRAM. To make maximum use of the 32 bytes of storage, the data is encoded in a compact form. In order to be useful to the program, it must be translated. The information stored includes terminal attributes such as baud rate and parity; radio-link attributes such as baud rate, packet length, and transmitter keyup time; display features such as case conversion, echo mode, auto line feed, screen width, and nulls required. Special command characters for flow control, exit to command mode, and editing features are also stored, along with the station call sign. These parameters are changed by commands to HLR, which maintains the compressed copy of the parameters by calling LLR whenever a parameter is changed. This copy is overlaid on the nonvolatile storage as volatile RAM data, and becomes permanent when the "STORE" line is toggled. The nonvolatile RAM is controlled through the parallel I/O port of the 6522 VIA.

Calibration Routines

The hardware design of the TAPR TPJC provides for on-board calibration routines. Jumpers are connected which allow the VIA Timer 2 to count the modulator or demodulator frequency to be calibrated. The VIA timers are reconfigured so that Timer 1 acts as a free-running count-down timer, counting at the 921.6 kHz system clock rate, and Timer 2 generates an interrupt after two cycles of the tone frequency being calibrated. By starting the timers simultaneously, Timer 1 can be used to count clock pulses for the duration of two periods of the frequency being calibrated. Two of the LED indicators are controlled in parallel with the microphone audio and HDLC reset lines, and are used in this routine as visual indicators of frequency deviation from the desired values. The timer routines in the interrupt dispatch table are replaced with special calibration service routines.

High-level Interface

Most of the entry points provided for HLR are concerned with buffer management. The I/O functions performed through these routines are: read a string from terminal or packet input buffer to a specified location; write a string from a specified location to terminal or packet output buffer; return character count for an outgoing packet in the terminal input buffer; update terminal input buffer; and return space-available count for output buffers. Other functions provided are: update nonvolatile RAM, temporary or permanent mode; enter calibration routine; force CW ID; and perform a soft reset. A special routine is called by HLR to notify LLR of a change of mode from "command mode" to "converse mode" or "transparent mode." All routines are called with the addresses of the arguments in processor registers D, X, Y, and U as needed.

Low-level Debug Program

A debugging facility is provided in the LLR, primarily as a software development tool. This program is invoked by a special user-settable character. In order to preserve as much information as possible about the system prior to entering the debugger, the active terminal input and output buffers are "frozen" upon entry by swapping the buffer pointers with a set of pointers used exclusively in the debugging program. All terminal input thenceforth is interpreted as commands to the debugger. These commands permit examina-

tion or modification of any addressable location, modification of the processor registers as they were upon entry to the program, or transfer of execution to any location. This allows test routines to be stored in RAM and executed. The user also has direct access to I/O addresses, and can observe the contents of I/O buffers without interfering with them.

DESIGNING THE TAPR TNC AUDIO INPUT FILTER

Margaret Morrison, KV7D and Dan Morrison, KV7B
4301 E. Holmes Street
Tucson, Arizona 85711
602-327-4775

Standard modulation for present terminal node controllers is Bell 202 compatible 1200 Hz / 2200 Hz phase-continuous FSK. Fig. 1 shows the typical spectral characteristics of such data. Notice that frequencies ranging from about 500 Hz to 2900 Hz are present in random NRZI data. One might guess that frequencies outside the central region from 1000 Hz to 2400 Hz could be eliminated with no degradation of demodulator performance. This turns out to be incorrect: the demodulator PLL needs this information to ensure timely response to data transitions. Thus, the ideal audio response over the link should be flat from below 500 Hz to over 2900 Hz.

In fact, the audio response of a typical FM 2-meter link looks like Fig. 2. This response curve shows dramatic rolloff over the frequency range of interest. Most of the rolloff is due to receiver audio characteristics and, in fact, some transmitted signals actually seem more like phase-modulation rather than FM, with high frequency emphasis. In many cases, the demodulator on the TAPR TNC cannot handle this rolloff, and successful demodulation of these signals requires reduced baud rates. We decided to put in a filter ahead of the demodulator to alleviate this problem.

The filter needed to take up as little space as possible, to be simple to modify, and most importantly, to be effective. In view of the fact that a variety of clock signals were already available on the TAPR TX, we opted for a switched-capacitor filter design, since ease of modification and parts-count make this a clear winner in the design competition. We quickly realized that in order t0 produce frequency compensation over a very wide range at least two sections would be required. Lyle Johnson, our hardware guru, thus selected the National Semiconductor MF-10 dual section filter as the part of choice, leaving the design details up to us.

Filter Configuration

This marvelous part may be configured in a number of ways as first or second order filter sections depending on external components, and each section may be configured independently of the other. After some experimentation we chose two second order sections -- one set up as a highpass filter, the other as a lowpass filter. This choice optimizes both amplitude and phase response over the wide range of frequencies required.

The manufacturer's literature lists nine different recommended configurations, each yielding a variety of output options. The choice is quickly narrowed to one, mode 3, which provides both highpass and lowpass outputs, and which provides maximum flexibility in selecting filter parameters. In particular, the frequency parameter , f0, is adjustable via resistor choices rather than by clock frequency alone. This, as well as dynamic range considerations, directed us to select mode 3 for both sections. In addition, we chose the option which scales f0 to fclk/100 rather than to fclk/50 in order to have as high a clock rate as possible. This allows the switching noise, which occurs at the clock frequency, to be most easily removed by post-filtering.

Filter Response

The input section, configured as a highpass filter, is shown in Fig. 3a; the second section is configured as a lowpass filter, shown in Fig. 3b. The frequency response for these two sections may be expressed as functions of frequency, f, and the parameters f0, Q, and either highpass gain parameter GLP or lowpass gain parameter GHP. Let $\omega = jf/f0$, where $j^2 = -1$. Then, the highpass filter's (complex) frequency response is

$$h(f) = GHP \frac{\omega^2}{\omega^2 + \omega/Q + 1} \quad (1)$$

and the lowpass response is

$$h(f) = GLP \frac{1}{\omega^2 + \omega/Q + 1} \quad (2)$$

For either, the amplitude response ma; be determined as $|h|$, and the phase response as

$$\phi_{out} - \phi_{in} = \arctan\left[Im(h)/Re(h)\right] \quad (3)$$

The parameters f0H, QH, and f0L, QL are related to the clock frequency, and the programming resistors, R2, R3, and R4, by the following equations:

$$f0 = \frac{fclk}{100}\sqrt{\frac{R2}{R4}} \quad (4)$$

$$Q = \frac{R3}{R2}\sqrt{\frac{R2}{R4}} \quad (5)$$

where F0 is f0H or f0L and Q is QH or QL, depending on the section.

For the highpass section, the gain is GHP = -R2/R1, while for the lowpass section the gain is GLP = -R4/R1. Notice that for either

section the input resistor, R1, only affects the overall gain and does not influence the shape of the response.

Optimization Procedure

The design task begins by choosing f0H, QH, f0L, and QL so as to produce as flat an overall filtered response as possible given the unfiltered response. These four parameters are determined by successive interactive computer optimization of the filtered response. A systematic search is made for the optimum parameters for each section alternately.

The optimization of either section consists of a nested iteration which determines its Q in an inner loop and its f0 in an outer loop. The other section's parameters remain fixed during this iteration. The inner loop calculates a least-squares straight line fit to the filtered audio amplitude response (expressed in dB), over a specified range of frequencies, and varies Q to produce a flat response., that is, best-fit slope of zero.

This procedure is written as a function whose value is the RMS deviation of the filtered response from the fit, and which also returns to the outer loop as arguments the value of Q, the slope, and the intercept of the fit. The outer procedure consists of a non-linear optimization (minimization) of the RNS error between the filtered response and the best-fitting frequency-indendent response, as a function of the f0 parameter.

The specified frequency range for the calculation can be varied to achieve best results. Typically, optimizing over the range 1000 Hz to 2400 Hz produces very good results, as rolloff begins well outside the range of optimization. It is helpful to be able to graphically display intermediate response curves during the optimization and to allow only inner-loop iterations on occasion.

Programming the Filter

The remaining work is to determine the eight resistor values (four in each section) which best implement the optimized response functions. Several considerations allow a unique choice to be made. First, for best dynamic range in mode 3, fclk/100 should be as near to f0 as possible. Since both sections run off the same clock, the optimum compromise clock frequency is the geometric mean,

$$\frac{fclk}{100} = \sqrt{f0H \cdot f0L} \qquad (6)$$

In practice, on the TAPR TNC this meant picking the nearest available frequency to this (fclk = 115.2 kHz), since the main crystal clock frequency is a power of two times any of the available values. Having selected the proper clock frequency we are able to determine the resistance values. This is done in the following steps:

1. Notice that in either section the input resistor, R1, does not determine response shape -- only overall gain of the stage. Using Eqs. (4) and (5) determine the ratios R2 : R3 : R4 which affect the shape of the response,

2. Temporarily design each section so as to have unity amplitude response at 1200 Hz. Having thus determined the overall gain parameters GLP and GHP, determine the ratio of R1 to R2 for the highpass section and the ratio of RL to R4 for the lowpass section based on the gain-resistor relationships.

3. Arbitrarily set the lowest resistance value in each section to 10 kilohms. This allows at least two of the eight resistors to be of garden variety. In addition, the resistances will be as low as possible while staying well above the minimum value recommended by the manufacturer (5 kil-ohms). Finish the design by using the ratios Ri : R2 : R3 : R4 to determine the remaining resistors.

4. (Optional) To further ease the task of obtaining resistor values, set both input resistors to 10 kilohms as well. Our experience is that for the filters we are designing, this has little effect on either the dynamic range or the overall gain.

The method outlined here efficiently produces broadband designs. The result of applying this procedure to the unfiltered response shown in Fig. 2 is shown in Fig. 4. The amplitude and phase response of this filter is shown in Figs. 5a and 5b. Using optional step 4 above, the overall gain of the highpass section is 6.73 and that of the lowpass section is 3.73. The resistance values, in kilohms, are:

Highpass	Lowpass
R1 = 10.0	R1 = 10.0
R2 = 63.7	R2 = 16.3
R3 = 58.3	R3 = 10.0
R4 = 10.0	R4 = 37.3

A single FORTRAN program was written to perform the design steps, menu driven and producing graphical display of the response as the design proceeds. A listing is available for the cost of photocopying, and the source is available provided a 9-track tape is sent to the authors..

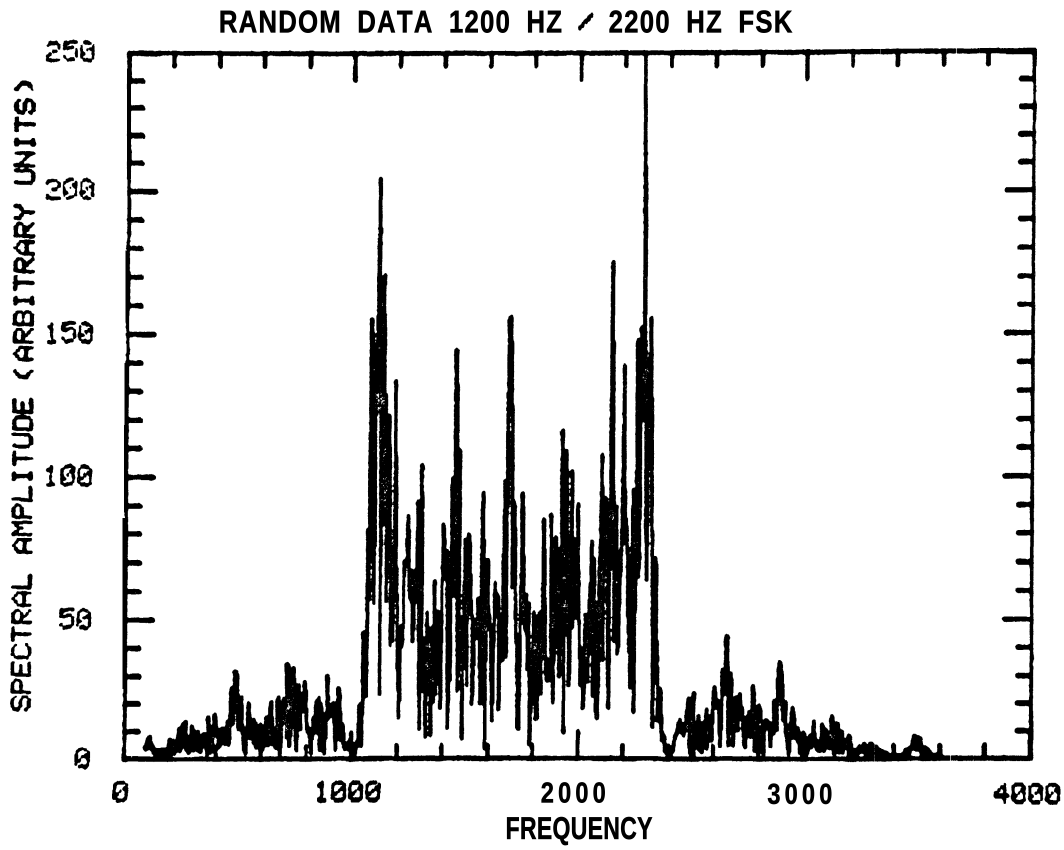


Figure 1. Typical spectral characteristics of random Bell 202 compatible phase-continuous FSK data.

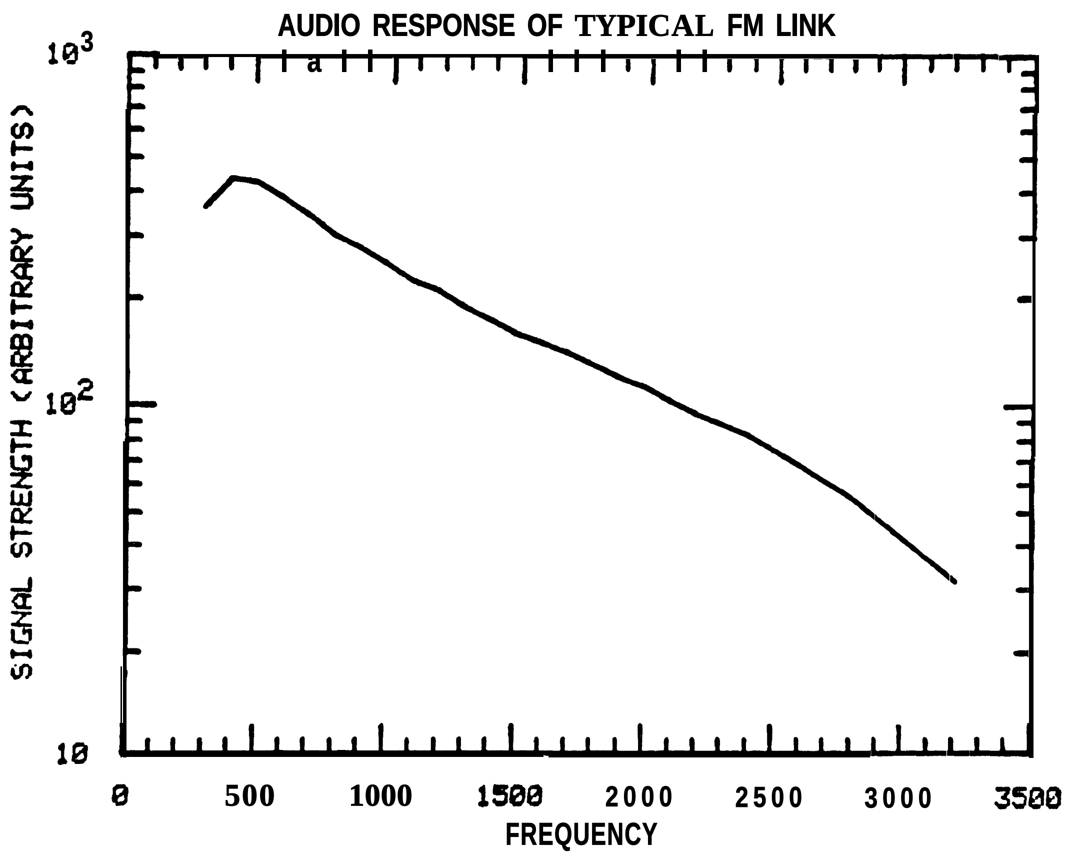


Figure 2. Audio response of a typical 2-meter FM link.

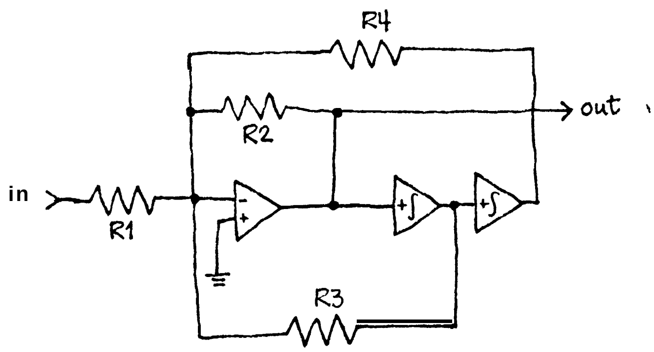


Figure 3a. Equivalent circuit for the input highpass filter section.

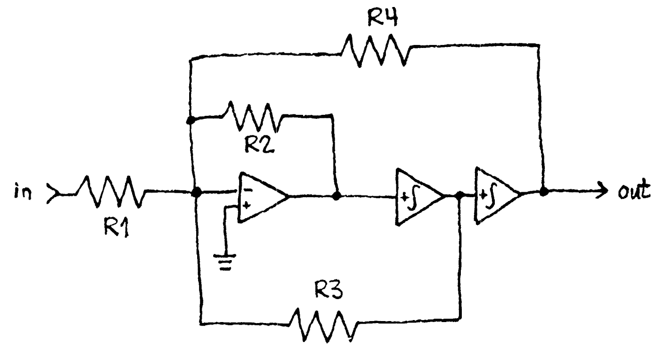


Figure 3b. Equivalent circuit for the output lowpass filter section.

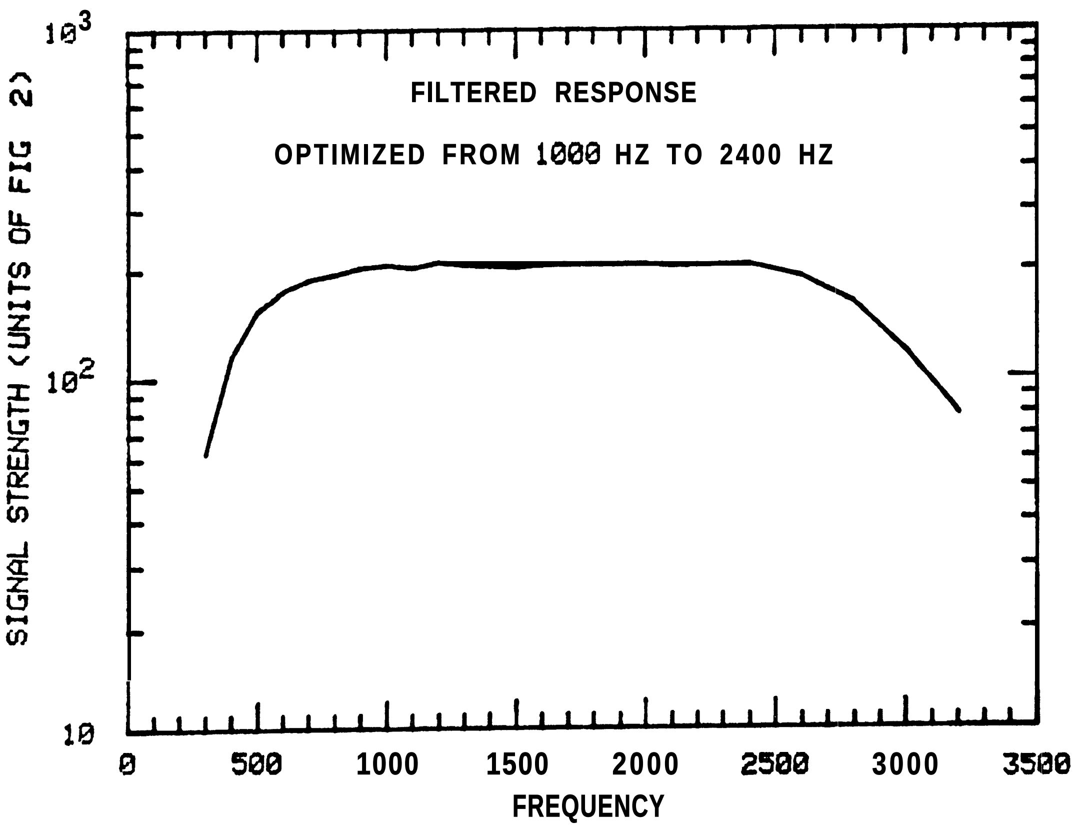


Figure 4. Filtered audio response of Fig. 2

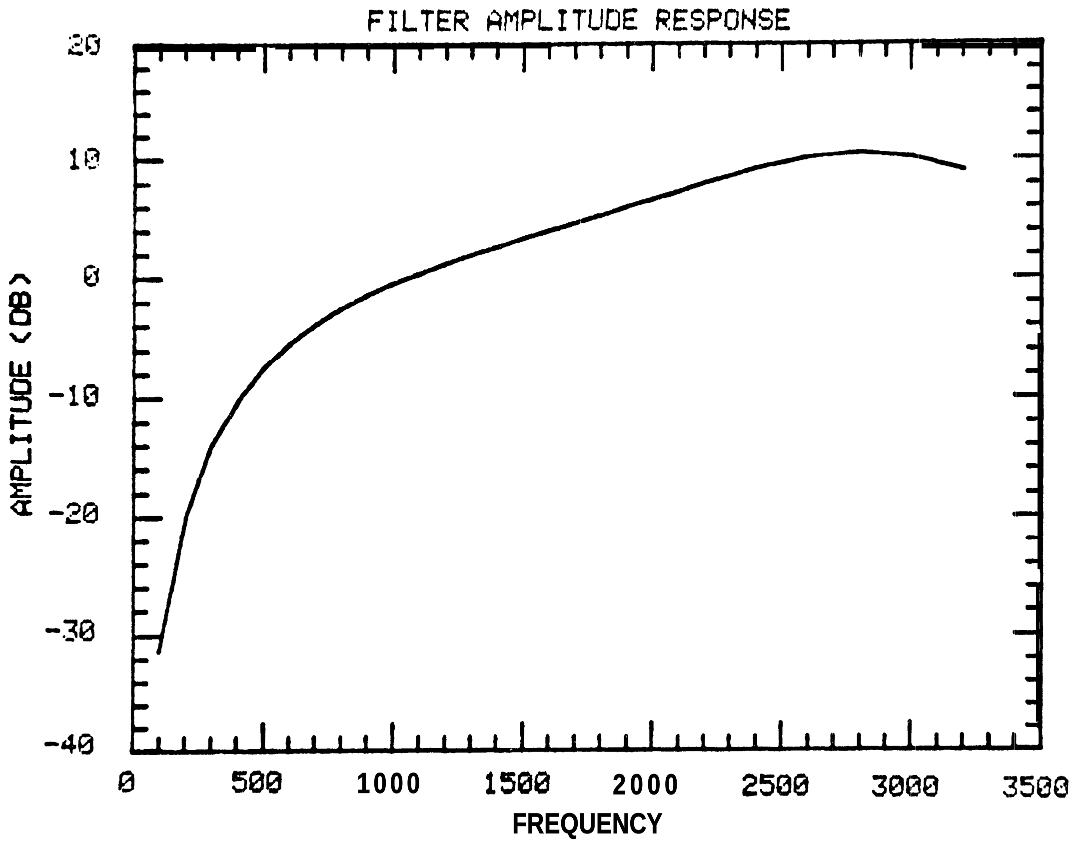


Figure 5a. Filter amplitude response.

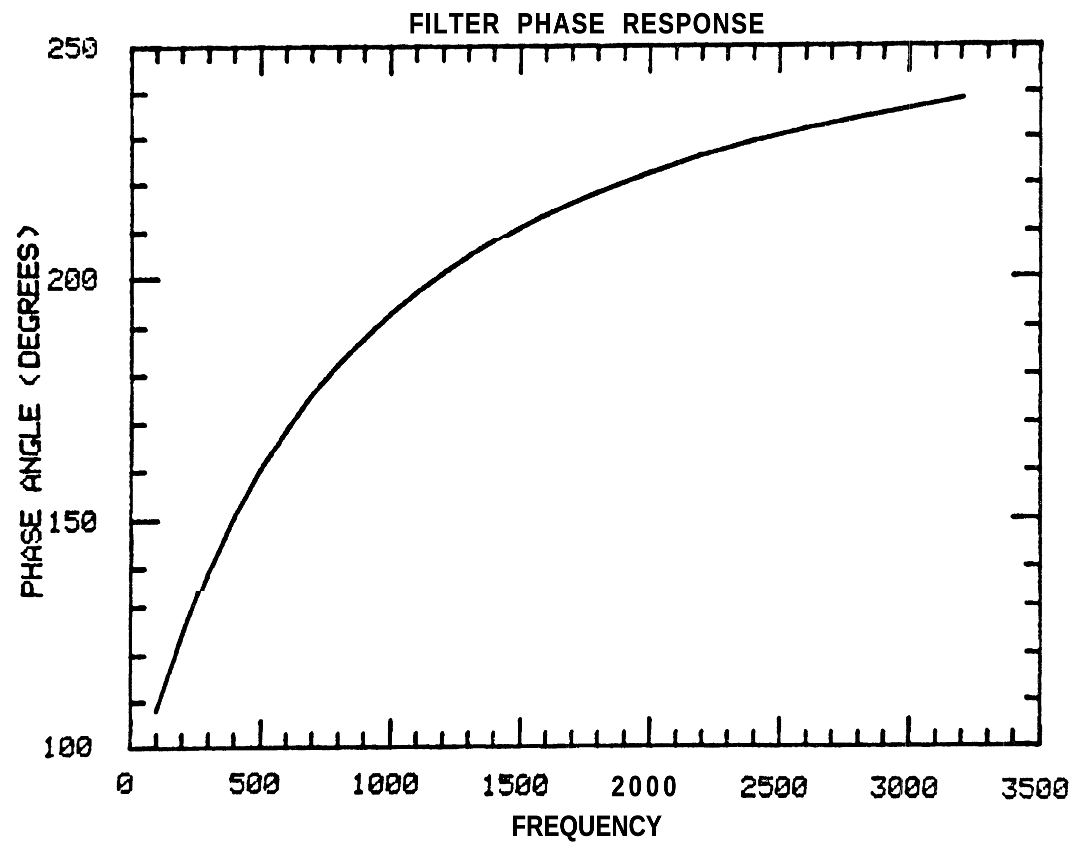


Figure 5b. Filter phase response.

PACKET RADIO FOR EMERGENCY COMMUNICATIONS

Bob Neben, K9BL
126 E. Schantz Ave.
Dayton, Ohio 45409

There is a need to redesign the techniques we use to handle emergency traffic. Many of us are combining processor controlled equipment and traffic handling techniques designed in the 1930's.

Traffic handling originated in radio, using CW, a continuance from the landline systems, I presume. This limits our copy to about 15 to 25 words per minute, depending upon the operator's ability. The reliability of this system is very good since a CW signal can punch its way through a lot of QRM and QRN. Accuracy, however, is limited to the accuracy of the sending operator and the receiving operator, both of whom are subject to fatique.

SSB or FM adds a new dimension, though, and we can talk about 150 to 200 words per minute. At these speeds, however, QRM is more of a problem. Also, we cannot pass traffic at that speed. Assuming we have to write the traffic on a message form, our speed decreases to about 25 words per minute, and we are really not much more ahead of the process than we were with CW. I remember copying MARS (Military Affiliate Radio System) traffic, and whenever possible, checking the addressee in the telephone book. It seemed more often than not that at least one digit was wrong.

RTTY automates what we were doing manually at speeds of 60 to 100 words per minute. Reliability is about the same as voice, and accuracy is slightly better. Maintaining good accuracy requires careful tuning, listening for a "hit", and human attention while typing. Generally I felt the accuracy of our MARS traffic left a lot to be desired.

The type of traffic influences both speed and accuracy. Ragchewing requires neither speed, accuracy, or hardcopy. "Traffic", such as health, welfare, or greeting messages is different. Any media or system we use has a maximum capacity. For instance, suppose we are passing messages using 100 words per minute RTTY, with no QRM, by continuously feeding paper tape to our TD (Transmitter Distributor). The system capacity would approach 100 words per minute in this case, and accuracy of the system would be very good. The dashed line (Figure 1) represents our system capacity. If we are using 60 words per minute RTTY, voice or CW, the dashed line would represent a different system capacity. Equally important though, is the type of traffic.

Normal day-to-day message traffic such as MARS, demands only a small percentage of system capacity. Even at peak periods such as holiday traffic, it can normally be handled during the alloted time for the traffic net. In Figure 1, traffic supplied equals traffic demanded, which is still below system capacity. System accuracy is fairly good since there is time for retransmission requests and no one is under any particular pressure,

Special events such as weather nets or public service events are difficult, as the traffic is not constant. System capacity is still constraining us, and the traffic demanded begins, reaches a peak, and tapers off, (Figure 2). In the case of a weather watch, t'here is a scramble to get the watchers in position. Traffic builds as the NWS (National Weather Service), EOC (Emergency Operating Center), or whomever we are assisting demands more information. usually, just about the time information is most critical, such as when the storm is directly overhead, the system becomes overloaded, and traffic demands have exceeded capacity. What happens? Well, if the net control. can keep a cool head and the net is well disciplined, some of the more routine traffic becomes delayed. Accuracy decreases, however, and sorting priorities becomes a problem. Is the Mayor's "routine" acted upon before the NWS "priority"? In time the delayed traffic is transmitted, but some of it will disappear, because it is no longer timely. This is not to imply it wasn't important, it was important, but we missed our chance. Somehow we need a better way of conducting traffic nets.

Disaster nets have a terrible efficiency, (Figure 3). The traffic demands build to gargantuan proportions following tornado touchdowns, and other major events. We work our system to capacity, but it takes days and even weeks to chip away at the workload. Accuracy is horrible, and faith in the system and 'amateur radio suffers in the long run. I could justify this scenario in the 1930's, but what do we answer in the computer age?

The answer to this problem is to move that system capacity line up so high that we couldn't run into it if we tried and at the same time do error checking to insure 100% system accuracy. This is exactly what Packet Radio will do for us in the amateur community, and it will do this at a relatively low cost.

A Packet Radio station consists of your present rig (1930 vintage if you so desire, but preferably a modern FM transceiver), some kind of terminal or personal computer, and a TNC (terminal node controller), which does the packet formatting, error checking, and several other functions. Computers are becoming available for $100.00 and up, and TNCs, such as the one offered by the Tuscon Amateur Pack Radio group (TAPR), sell in the $200.00 range. So the cost to upgrade your station to Packet Radio is perhaps the cost of a two meter rig.

Packet Radio will do a number of things for us. It will change the system capacity line from 100 words per minute in our example (74 Baud) to 1200 Baud. On paper that's a sixteen fold increase. In reality, it will be less because of packet overhead, but the increase is still phenomenal. The accuracy is virtually 100%, because of error checking and system acknowledgements. Previously, the net controls could only talk to one station at a time. In Packet Radio, the 64 stations in each local area network (LAN) can send data to other stations simultaneously.

Computers don't have much effect on our present traffic systems, since human intervention is usually required to check status, stored messages, etc. In Packet Radio, we have many uses for the computer. We could store messages for a station not yet logged in. We could do inquiries, such as health and welfare traffic. This may best be done computer to computer, which is fairly easy to set up. We could tie our computer to others in the area over land line, or another frequency to handle incoming traffic. The possible uses for our home and club computers is seemingly endless.

Our traffic nets are usually single function; VHF for local, HF for large areas, etc. By using the gateway function, our LAN Packet System can access worldwide via satellite. This provides a means to get traffic in and out of the local system. Perhaps we need local stations to handle the LAN. The other four stations (or more) could link to other LAN's, gateways, computers, etc.

What could happen if the national emergency evacuation plan were implemented? Imagine moving 100,000 people in your community to an area 50 miles away. It is logical that amateur radio would be used to help coordinate this massive effort. How would we handle this? The logistics would involve massive vehicle movement, fuel, food, medical care, etc. Our present system would have little usefulness, but a Packet Radio system could easily accommodate this. If one LAN becomes overloaded, just initiate another. The gateways would also be heavily used and again, if a gateway becomes overloaded even at 19.4 K baud, another gateway would be initiated.

We are still using old technological equipment. Old technological traffic handling techniques are effective for day-to-day operation, but become overloaded at the first sign of large scale activity. We have the technology to correct the situation, but we need to act now to adapt Packet Radio technology and procedures to our traffic handling system.

System Capacity
Traffic Demands
Traffic Supplied

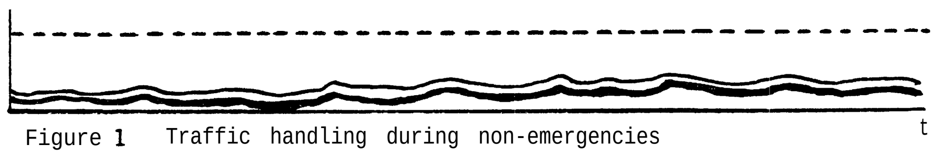


Figure 1 Traffic handling during non-emergencies

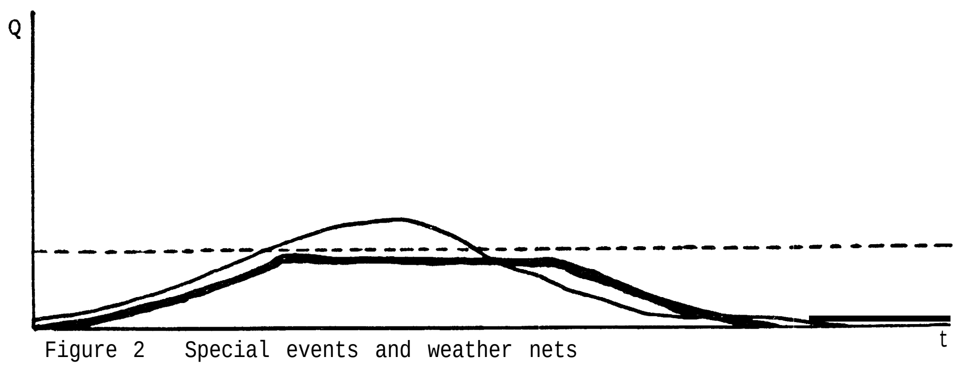


Figure 2 Special events and weather nets

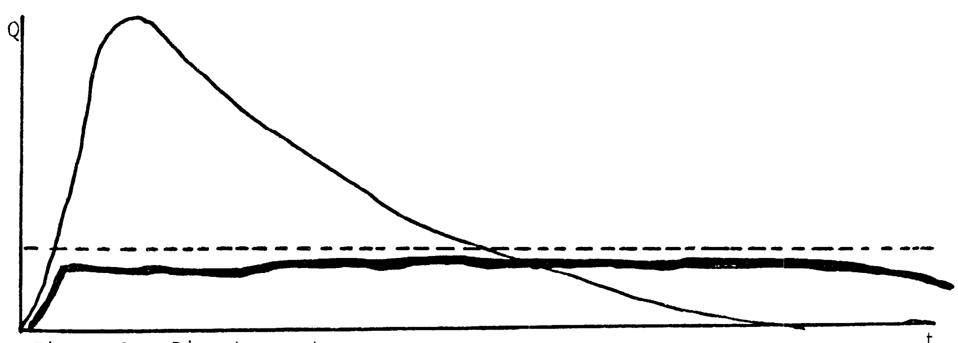


Figure 3 Disaster nets

Traffic Activity Chart

MULTI--USE DESIGN CONSIDERATIONS FOR THE TAPR TNC

Harold E. Price, NK6K
2110 Farrell Ave #14
Redondo Beach, CA 90278

Abstract

The Amateur Packet Radio Terminal Node Controller (TNC) built by TAPR Inc. is designed to meet the needs of a wide range of users, from the technical experimenter to the "appliance operator" end user. This paper discusses the human engineering factors which went into the design of the TNC's external software interfaces that enable it to serve a heterogeneous user base.

Background

The first Terminal Node Controller widely distributed for amateur use was produced by the Vancouver Area Digital Communications Group, It was mainly distributed in bare-board form, and aside from the specialized HDLC communications chip, was able to be stuffed with readily available components. Once the user supplied a modem and power supply he had a device which required only a terminal and a radio to get on the air and send packets. The capability supplied by the VADCG TNC to newcomers in the field of packet radio far surpassed anything else available at the time, and has earned it its place in the Amateur Radio Hall of Fame.

The TAPR TNC was designed to go beyond the capabilities offered by the previous TNCs. Taking advantage of advances in technology, including the reduction in price of denser memories, the TNC hardware offered the following features:

- 24K ROM
- Non-volatile RAM
- 6K RAM
- Onboard power supply with an off board transformer included.
- Onboard modem
- The TNC would be offered assembled and tested.

The increased program space made possible increased software functionality. The increased level of pre-packaging would make the board attractive to an additional class of users, the "appliance operators". This term applies to that group of people who buy a product and expect to plug it in and use it. The term is sometimes used in a derogatory sense but that is not the intent here. This class of TNC user is not interested in the underlying concepts and does not wish to tinker with the innards of the TNC. They accept packet radio as a proven technology and want to get on with the business of being end users, setting up higher level networks, establishing mailbox systems, building local area nets with intelligent host nodes and file servers, or simply using the TNC as error free RTTY. The TMC is a secondary item, simply a means to an end.

This type of user would not be happy if he was required to modify source code and recompile to change node address, call signs, and other operational parameters, On the other end of the spectrum is the experimenter, those individuals interested in tinkering with the lower levels. Many areas of amateur packet radio remain open issues, What are the best timing algorithms? What are the correct retry counts, timeout values, keyup delays, packet lengths and other details associated with the delivery of HDLC frames from one TNC to another? The functions offered by the TAPR TNC to solve these basic compatibility problems are discussed in the following sections.

Conflicting Goals

To illustrate the types of problems involved, the following items are presented, each with its set of conflicting viewpoints.

Degree of Transparency

In applications where the TNC is treated as a means for connecting two devices, computer to computer or remote terminal to computer, the TNC should be totally invisible. It can be viewed as a simple modem, receiving characters from data terminal equipment (DTE) and sending it down a phone line, and receiving data from the line and sending it to the DTE. The TNC has no personality of its own,

In other applications, the TNC is itself the smart device at the end of a communications link, using a terminal only as an IO device under its control. The TNC performs local input editing and output formatting. It attempts to preserve visual fidelity on the terminal device, i.e., not mixing the echo of data being entered for transmission with data received from the link,

Command set

Infrequent (casual) users require a simple command set with straightforward syntax and easy-to-remember english-like commands. Changing TNC major operating modes should require a small number (one) of commands.

Expert users or those experimenting with operational parameters require a rich command set. This class of user also prefers commands needing only a small number of keystrokes as the command entry rate is higher.

Tuning Knobs

This is closely related to command set, in that the more things there are to adjust, the more commands will be required to adjust them.

To use a radio analogy, some hams are happy with ON/OFF, push to talk, and frequency tuning, others require variable bandwidth, IF shift (both high side and low side), notch, RF attenuation, multiple tuning rates, ten VFOs and snooze alarm.

In addition, experimenters want access to all parameters which control forming, sending, and receiving packets.

These are examples of the type of conflicting interests generated by a diverse user base. Availability of a 24K program space on the TNC coupled with the means to develop software in a high level language (Pascal) permitted the three-person software team to provide a single integrated software package which meets everyone's needs. The major features of this design are discussed below.

Definition of Terms

Several of the shorthand terms used in the remainder of this paper are defined here.

User interface. The asynchronous data path to the TNC, the path used to exchange data which is formatted and transmitted over the air. Configuration commands are also given to the TNC through this path.

Link interface. The bit stream (HDLC) path which connects directly to the RF transceiver. Rackets (frames) are sent over this path.

Connected. A TNC is "connected" when it has exchanged the proper sequence of packets over the link interface with another TNC such that the two TNCs are connected at the AX.25 (or VADCG) level 2 layer. Data sent by the TNC will be in the form of sequenced (numbered) information frames. Connected does not refer to any physical connections.

Unconnected. The TNC is not "connected" to any other TNC. Data sent by it on the link is in the form of unsequenced information frames.

Data transfer mode. Characters received through the user interface are assumed to be data for eventual transmission through the link when the TNC is in a data transfer mode. When not in this mode data input is assumed to be a configuration command.

Operating modes

The major point of difference between users of the TNC is the degree of transparency of the user interface when the TNC is in a data transfer mode. Put more sirnply, is the TNC something one speaks TO or speaks THROUGH? As stated previously, sorne users want to treat the TNC as a smart terminal, others view it as a smart modem, to be heard but not seen, Because of the large number of differences in basic TNC functions required to satisfy these two needs, the TNC software supplies two data transfer modes. Entering one of these modes has the effect of changing the values of a large number of configuration parameters at once. The data transfer modes are the CONVERSATIONAL mode and the TRANSPARENT mode, and are abbreviated to "convers" and "trans" .

Changing modes causes a radical change in behavior of the interface. Attributes of the TNC in CONVERS mode are:

1) Characters input are scanned for special meaning to implement f'eatures discussed below.

2) Creation of packets is directly under user control. A special character is used to say "take the data entered previously, make it into a packet, and send it as soon as possible".

3) Data can be edited until released to be made into a packet. Characters can be struck out, a line erased, or the entire packet can be erased.

4) Data input is echoed locally, i.e., by the TNC.

5) Data received from the link is formatted before being output.

6) Flow control is available via use of XON,'XOFF characters. Flow control is the process where data buffer or CRT display overflow is avoided by stopping the flow of data, The XOFF character is a request to halt the flow of data, XON is permission to resume t'he flow through the user interface. Case translation is performed, line feeds can be inserted in the data stream following carriage returns, carriage returns are inserted when the output line length is exceeded.

The attributes of the TNC when in the TRANS mode are:

1) All characters are transferred to the link without modification.

2) Creation of packets is only under the indirect control of the user. Packets are formed by time or by data length. Data length control causes packets to be formed after "n" characters have been entered through the user interface. Time control will build a packet every "n" seconds or after "n" seconds have passed since the last character was received from the user interface.

3) Local echoing and local editing are not available.

4) Data received through the link is transferred to the user interface as received, without modification or addition of control characters.

5) Flow control is not controlled by data characters, but is managed by use of the standard RS-232 handshake lines.

These two data transfer modes allow the TNC to be compatible with a wide range of applications. The behaviors described are present by default, they may be modified at will by the user. Functions can be added to TRANS mode, or removed from CONVERS mode, supplying a variety of hybrid semi-transparent or translucent modes.

A third operating mode is present in the TNC, COMMAND mode. Command mode is active when the TNC is first powered up, and can be entered from either of the other modes. COMMAND mode is used to change the configuration and operating parameters of the TNC. Changing the TNC from mode to mode does not effect the state of the link, i.e., whether or not the TNC is connected. This allows two personal computer owners to talk back and forth in CONVERS mode, and then enter TRANS mode to let their computers transfer files.

A problem arises with respect to the total transparent mode. Short of a hardware reset, how is the command mode entered from the transparent mode? The method used to leave convers mode and enter command mode, a special control character, can not be used. Instead, a small compromise is made. After a user defined period (guard time) has passed where no characters have been received from the user interface the TNC will watch for a mode escape character. If this character is entered three times with no intervening characters, and another guard interval passes with no additional input, the TNC enters the command mode. A proper selection of the guard time and escape character will provide a high probability that reception of the sequence is a valid request for command mode entry. This procedure did not originate with the TAPR TNC and is a common practice for intelligent modems.

Parameters

There are many facets of the user and link interfaces that experimenter will want to twiddle with and the end user will want to ignore, or at most change only once. Examples of user interface items are:

- Editing characters such as character delete, line delete, and packet delete.
- Flow control characters.
- Packet creation times in TRANS mode.
- Output presentation control such as <lf> after <cr>, number of nulls after <cr>, and terminal line length.

Examples of link control items are:

- Transmitter and repeater keyup delays.
- Number of times to retry a frame, frame acknowledge time
- Packet data size
- Maximum contiguous frames sent or maximum number of outstanding unacknowledged frames
- Station address or call sign.

Previous amateur packet systems kept these parameters in ROM and supplied no way to modify them at run-time. The TAPR TNC software embodies a design methodology where all such values are kept in RAM while the TNC is running, and therefore can be modified by user command. The values are initialized from ROM when the TNC is first installed. The large number of parameters available to the user (see table 1) would require a very tedious startup procedure every time power was applied as not all users would be happy with the default values supplied in ROM. To avoid this unpleasantness, the TAPR TNC uses non-volatile RAM to store parameters when the TNC is turned off. A dip switch on the TNC selects either the ROM copy or the non-volatile RAM copy for initialization of the RAM parameters, After an initial boot from ROM, the TNC is then configured to use the non-volatile RAM for subsequent booting.

This combination of software and hardware design permits hands-off operation for some users, one-time configuration for end-users, and endless opportunity for change by experimenters.

Commands

The desire to have many user accessible parameters conflicts with the desire to have a small number of simple commands. Since a major design goal was to externalize as many operational parameters as possible, it was soon realized that the TNC would have several commands. The Beta test release of the TNC software has 66 separate commands.

Only four commands, the commands to connect and disconnect the link and the mode change commands, are used in normal operation of the TNC. Most of the other commands are used to change parameters which have default values pre-assigned when the TNC is first powered on. These values are satisfactory for most applications and will never be changed. If the parameters are changed, however, the user can issue a command that causes the changed values to be stored in the non-volatile RAM, replacing the default values.

The command syntax is kept simple, consisting only of the name of the parameter to be changed and the new value. A DISPLAY command is present which lists all parameters and their values. While not a replacement for a HELP facility, the list of parameters names serves as a memory jogger and will cut down on trips through the manual.

To reduce the number of keystrokes needed, all commands can be abbreviated to a smaller number of characters that still uniquely identify the parameter,

Documentation

As much care was taken with the design and implementation of the manual as was taken with the design and implementation of the hardware and software. The documentation must serve the same diverse user base the that TNC does. Because of the need to provide both tutorial and detailed information, the manual for the TNC is approximately 140 pages long. The manual is structured to supply the right level of detail at the right time. The first few pages deal directly with the tasks required to get the TNC on the air, including a detailed radio interface section. This detail is required because incorrect wiring of the TNC to RF interface can result in damage to the TNC and/or the RF gear. The introductory material is keep as short as possible however, and by page 28 the user will be on the air sending packets.

Subsequent sections of the manual provide tutorial information on the hardware and on the protocols used. Other sections provide detailed information on the software command set and on the TNC hardware. A full set of schematics is provided for hardware experimenters. Also included in appendicies are the specifications for the communications protocols used on the TNC. These supply sufficient detail for others to implement compatible software.

The manual is structured to supply the level of detail required by experimenters while at the same time not scaring off the casual user.

Summary

Great care was taken during the design and implementat ion of the TNC so-ftware to provide a simple interface for end users while not limiting the activities of more advanced users. The TAPR TNC is currently in a Beta Test phase with 180 users spread out in more than 16 local groups. It is expected that new commands will be needed while some current commands will prove to be superfluous and can be removed. The large percentage of high level code and human engineered design should make this a relatively painless task.

Acknowledgements

Many thanks go to Margret Morrison, KV7D, and Dave Henderson, KD4NL, who along with the author designed and implemented the TAPR AX.25 software, and to Dan Morrison, KV7B, for his design inputs.

Command	Type	Function
ABAUD	U	Asynchronous 'baud rate
ABIT	U	Asynchronous stop bits
AUTOLF	U	Follow received <cr> with <lf>
AWLEN	U	Asynchronous word length
AX25	L	AX.25 or VADCG protocol
AXDELAY	L	Keyup delay time for audio repeater
AXHANG	L	Audio repeater dropout (hang) time
BEACON	L	Beacon mode
BKONDEL	U	Echo <bs> when <del> entered
BTEXT	L	Beacon text
CALIBRAT	C	Enter hardware calibration routine
CANLINE	U	Cancel line character
CANPAC	U	Cancel packet character
CMDTIME	U	Command mode entry from TRANS mode timer
COMMAND	U	Command entry from CONVERS mode character
CONMODE	D	Default data transfer mode
CONNECT	L	Connect TNC to another TNC
CONOK	L	Permits connections in unattended operation
CONVERS	D	Enter CONVERSation mode
CPACTIME	U	Create packets by time in CONVERS mode
CR	U	Append a <cr> to the end of each packet in CONVERS mode
DEBUG	C	Enter the debugger
DELETE	U	Specifies which of <del> or <bs> deletes characters
DIGIPEAT	L	Enables AX.25 digipeating function
DISC	L	Disconnects the TNC from the link
DISPLAY	C	Displays all configuration parameters
DWAIT	L	Digipeater wait time
ECHO	U	Local echo in not in transparent mode
ESCAPE	U	Character echoed when <esc> is entered
FLOW	U	I/O data flow in CONVERS mode
FRACK	L	Frame acknowledge timeout
HBAUD	L	HDLC (link) baud rate
ID	L	Send CW id
LCOK	U	Lower case translation
MAXFRAME	L	Maximum unacknowledged frames
MONALL	L	Link monitor command
MONCON	L	Link monitor command
MONFROM	L	Link monitor command
MONITOR	L	Link monitor command
MONT0	L	Link monitor command
MYCALL	L	Call sign (address) of this TNC in AX.25 mode
MYVADR	L	VADCG protocol address byte
NULLS	U	Number of nulls added after <cr> in CONVERS mode
PACLEN	L	Maximum packet data length
PACTIME	U	Packet creation time for TRANS mode
PARITY	U	Parity of asynchronous user interface
PASS	U	Send next character verbatim in CONVERS mode
PERM	C	Put current parameter setting in non-volatile RAM.
RESET	C	Software reset
RETRY	L	Frame retry count
SCREENL	U	Screen width for CONVERS auto-wrap feature
SENDPAC	U	Create packet character for CONVERS mode
START	U	Flow control character
STOP	U	Flow control character
TRACE	L	Link diagnostic command
TRANS	D	Enter TRANSparent mode
TXDELAY	L	Transmitter keyup delay
TXFLOW	U	Use character flow control in TRANS mode
UNPROTO	L	Unconnected address field
VDIGIPEA	L	Enables VADCG digipeater function
VRPT	L	Direct VADCG frames sen by this TNC to a digipeater
XFLOW	U	XON/OFF or hardware flow control in CONVERS mode
XMITOK	L	Enable transmit functions
XOFF	U	Flow control character
XON	U	Flow control character

L - Link commands U - User interface
C - TNC control D - Data transfer mode

Table 1. TNC commands

Packet Radio - A Software Approach

Robert M. Richardson, FJ4UCI-I
22 North Lake Drive
Chautauqua Lake, N.Y. 14722

Abstract:

A software rather than hardware approach to synchronous Packet Radio communication at 1200 or 2400 Baud using the Radio Shack TRS-80, Model I or Model III microcomputer is described. The program duplicates virtually all the functions provided by the Vancouver Area Digital Communications Group (VADCG) terminal node controller board which requires an 8085 microprocessor, an 8273 synchronous data link controller (SDLC), an 8250 serial I/O, and a number of EPROM and RAM memory chips, plus a separate microcomputer with RS232C interface and a 1200/2200 Hz modem.

The only external equipment required by the software approach, other than a TRS-80, amateur VHF transceiver and antenna, is a port zero encoder/decoder, and two EXAR chips for AFSK keying and demodulation. The program has been extensively tested on the 2 meter amateur band working into southern Ontario and locally in western New York.

Introduction:

Packet radio communications is coming down the amateur radio pike in the near future, and the near future is now upon us. We have closely followed the evolution of amateur packet radio with its asynchronous beginnings in the Montreal area during early 1979 and its synchronous beginnings in the Vancouver area later in 1979/1980. The synchronous packet radio pioneers, including Douglas Lockhart VE7APU et al in the Vancover area developed the rightly famous and widely used VADCG terminal node controller board which was the introduction to synchronous amateur packet radio for the majority of all amateurs actively participating in packet communications today.

Though a number of SDLC controllers in addition to the Intel 8273 are now available, such as those from Western Digital and Zilog, and the price will hopefully be coming down as volume increases, there is yet another approach to amateur packet radio which due to its low cost and simplicity, may greatly broaden amateur participation in this wave of the future. This approach is the software rather than firmware approach using a Model I or Model III TRS-80 with 48K memory and hopefully for convenience, 1 or 2 mini-disk drives. Suffice it to say, assembly language is used which offers nearly 300 times faster execution speed than standard Fortran, Pascal, or Basic high level languages.

Even with the remarkable speed assembly language offers the user, it does require a finite amount of time in the receive mode for the program to serially accomplish what the dedicated SDLC chips are able to accomplish in parallel: i.e., find the last openning flag and address and store the packet bits in memory, convert the serial packet bits to decimal with zero deletion (the opposite of zero insertion), and do the CRC checking for each frame. Nevertheless, this entire process for the average packet only requires 90 milliseconds which is not a significant or even noticeable time delay to the operator at either end of the circuit.

General:

The program is comprised of 5 segments:

1. The transmit mode segment which does the work horse job of converting prepared messages or programs, either keyboard or disk input, into IBM SDLC format, and clocking them out bit by bit serially at the desired Baud rate via port zero. Packet and frame length may be any length desired from 1 to 250 bytes plus address, control and CRC bytes, and may be input from the menu. See figure 1. A total packet or frame length of 256 bytes will be allowed when intra repeater routing comes to pass. The extra two bytes will serve to determine routing. The number of preamble flags sent before the packet may also be programmed which is a courtesy for those with slow transmit and receive switching at the receiving end, A number of prepared messages may selected from the menu as well as keyboard input to memory for a nearly immediate packet. The video display utilizes the split screen format with the top 8 lines for receive and the bottom 8 lines for transmit. See figure 2. Independent sequential scrolling is provided for both transmit and receive modes.

2. The cyclic redundancy check,

CRC16, segment which automatically generates the 2 byte CRC in IBM modulo 2 format and appends these 2 bytes to each frame transmitted. This same subroutine serves the receive mode to check each incoming frame For CPC validity also using the IBM modulo 2 CRC algorithm. which the dedicated SDLC chips utilize.

3. The receive mode segment has a software equivalent of a digital phase locked loop which takes the serial packet input via port zero, and stores each bit in memory either 8 bits per byte for best memory utilization, or 1 bit per byte for instructional purposes (easy to visualize). After the complete packet has been received, it is first converted from binary to decimal, then stored in high memory and displayed on video while the CRC test is being made for each frame if a multi frame packet was received. Depending upon whether the packet is of the unnumbered, supervisory, or information variety, appropriate action is taken automatically. The conventions followed are those used by the VADCG so the program is fully compatible with those amateurs using the Vancouver terminal node controller. The program will receive and decode packets of any length up to the 12,288 byte capacity of the bit store allocation. Average packet decoding time after reception of the packet transmmission is 90 milliseconds, as previously mentioned.

4. The edit/modify segment is a unique and extremely useful utility that allows the operator instant access to any 1 of the 1024 byte, 60 pages of memory in the TRS-80. Each page of memory is displayed at one time on the video display in TRS-80 ASCII or graphics format. Either upper case only, or upper and lower case edit/modify may be selected from the menu. Pressing the 'M' for modify key intiates a flashing cursor that may be moved up, down, left, or right on the displayed page with the arrow keys. Keyboard input ASCII value may be input directly if desired or pressing the shift '@' keys displays the memory location in decimal, memory value at this location, stack pointer value, and the operator asked to input the new value. Upon inputting the new value, the full memory page is displayed again, Pages are moved up or down through memory by pressinq the BREAK key (flashing cursor disappears) and then the ENTER key to move up a page in memory or the minus/DASH key to move down a paqe in memory. Needless to say, though ROM may be examined with this function, only RAM may be modified. To elininate tedious paging by hand, a number of control keys allows instant access to the more frequently used memory locations; i.e., transmit program store, received bit store, and received message store.

5. Morse code I.D. and Morse transmit segment. This minor subroutine called by shift 'I' sends the Morse I.D. of the transmitting station at any speed desired to satisfy F.C.C. requirements. Conversely, Morse code at any speed may be tranmitted from the keyboard via shift 'X'.

Memory Management:

The operating program resides in memory from 29696 through 40959. Three 12,288 byte segments are used exclusively by:

A. 17408-29695 is reserved for the program or data to be transmitted store. Normally, a disk program to be transmitted via packet is first loaded from disk to 40960+ in memory and then moved down to the 17408+ area by pressing shift 'Y'. In the 'connected' mode of operation the program or data fron 17408+ is automatically transmitted in single frame packets of 254 bytes length (a western New York convention so that a single station may NOT monopolize the packet repeater unfairly). This automatic transmission is called by pressing 'B' from the menu. Acknowledgement (ACK) packets are received automatically and if valid the next packet automatically transmitted, otherwise the previous packet is retransmitted.

B. 40960-53247 is reserved for incoming received bit storage. We prefer to store a bit per byte so that the user may visualize the stored bit pattern, though 8 bits per byte could be used for better memory utilization. Unless directed by the menu command to SAVE bit storage, this area is automatically cleared after each packet is converted to decimal and stored in high memory. Surely all you amateurs operating VADCG terminal node controllers know that it sends two SDLC logical zero 'bytes' AFTER the opening flags, which are then followed by a single flag BEFORE the address byte. Or did you? Packets less than 4 bytes total length, are ignored by the SDLC protocol that is used in this program.

C. 53248-65535 is utilized for storing the converted decimal byte values from received packets. When it is full, an automatic 'not ready to receive' (RNR) = 'wait' is transmitted and when acknowledged, the operator may clear out all address, control, and CRC bytes by pressing '$' from the menu. Pressing shift 'B' then takes the operator to DOS ready where the received program or data may be DUMPed to disk, after which one returns to the program, sends a 'ready to receive' (RR) to 'clear' the previous 'wait' and continues upward and onward if it is indeed a program or data longer than 12,288 bytes bytes in length.

Conclusion:

Amateur radio has room for all types and varieties of individuals with

dramatically differing interests. Some prefer operating and could care less how an electron gets from 'A' location to 'B' location and what is involved in getting it there, There is certainly nothing wrong with that, and to that variety of amateur we highly recommend the Vancouver terminal node controller kit *. With a few hours of soldering you are 'on the air' using their EPROMs. To the variety of amateur who truly wishes to thoroughly understand how the absolutely brilliant IBM SDLC protocol operates, how to modify it for the forthcoming HDLC modes, we recommend the software rather than hardware approach. When you are done, you will not only be an 'instant SDLC expert,' but a better assembly language programmer too.

The software approach to packet radio communications is obviously limited by the clock speed of the microcomputer being used. The Model I and Model III TPS-80 will easily handle 1200 and 2400 Baud packets, with 4800 Baud packets being somewhat marginal but acceptable, without modifying the standard crystal clock of either microcomputer. By installing one of the numerous clock speed up kits available (4 MHz), both 4800 and 9600 Baud packets could be handled.

We are indebted to a number of our Canadian neighbors in the Hamilton and Toronto area for assistance in testing the software approach to packet radio. Most noteable have been VE3MWM, VE3DSP, VE3IUV, and VE3DVV. Their packet repeater located in the southern environs of Toronto with the apt call sign of VE3RPT is now active on 145.650 MHz.

Locally (6.5 miles northeast of our QTH), the major effort to install a packet repeater in the greater Bufffalo area has been borne by W2EUP with considerable assistance from VE 3MWM. It will be linked to both the Toronto repeater to the north and a new packet repeater in the Syracuse area that will be able to link into the greater New York City area to the east, and thence linked up and down the east coast.

We have only touched briefly on a few of the highlights of the software approach to packet radio communications. To go through it in depth would take much more than the allotted space and time. For those who wish to explore the subject in greater depth, at least a few hundred pages worth, we suggest you watch for the new book, 'The Packet Radio Handbook' that will be published spring '83 by Richcraft Engineering Ltd., #1 Wahmeda Industrial Park, Chautauqua, N.Y. 14722 at $22 postpaid to the U.S. and Canada and $30 postpaid overseas.

* Vancouver Area Digital Comm. Group
818 Rondeau Street
Coquitlam, British Columbia
Canada V3J 523

```
                    ENTER OPTION DESIRED ? _

CHANGE ADDRESSEE & NUMBER = A     W2EUP CONNECT REQUEST CQ     = B
NOT CONNECTED TOGGLE      = C     W2EUP DISCONNECT REQUEST     = D
SEND PACKETS FROM LO-MEM  = E     W2EUP CONNECT ACKNOWLEDGE    = F
USING ONLY SOFTWARE MSG   = G     WORKING ON AMSAT AX, 25 MSG  = H
NOT INSERT DPLL BIT TOGGLE = I    SEND HI-MEM CONTINUOUSLY     = J
NOW IN UFPER CASE MODIFY  = K     FILL HI-MEM WITH UUUUU       = L
DISPLAY/EDIT MEMORY PAGE  = M     SET DISCRETE PACKET LENGTH   = N
LOAD HI-MEM ALL 11111     = O     CHANGE DPLL TIMING VALUE:    = P
SEND CONTINUCUS FLAGS/126 = Q     ABORT LO-MEM XMIT SEQUENCE   = R
MULTI-FRAME PACKET TEST   = S     QUICK BROWN FOX TEST MSG     = T
CLEAR NON-PGM MEM 17K-65K = u     INPUT/TRANSMIT MESSAGE = V & W
ANY PACKETEERS ABOOT 255/0 = X    MOVE MID-MEM TO LO-MEM       = Y
NORM BIT STASH CLEAR      = 1     MOVE RECV PACKS TO LO-MEM    = 2
SET NUMBER OPENING FLAGS  = 3     WAIT & CLEAR WAIT TOGGLE     = 4
```

Figure 1

```
1200 BAUD SDLC RECEIVE MODE                 ----> NOW CONNECTED
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1200 BAUD SDLC TRANSMIT MODE                 CONNECTED TO VE3MWM
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

Figure 2

INTRODUCING THE PACKET ADAPTIVE MODEM (PAM)

Paul L. Rinaldo., W4RI
Amateur Radio Research and Development Corp. (AMRAD)
P.O. Drawer 6128
McLean, Virginia USA 22106

Abstract

This paper describes a modem design undertaken by Robert E. Watson and the author. The modem was designed primarily for high-frequency packet radio applications. It operates at signaling rates of 75, 150, 300, 600, and 1200 bauds. The data rate is software controllable through a modified RS-232-C port. A frequency shift of 600 Hz is maintained for all data rates. The modulator is phase continuous and provides X32 or X64 clock to the packet assembler/disassembler (PAD) or terminal-node controller (TNC). The demodulator employs a National MF10 switched-capacitor filter (SCF) chip for each of the 1500-Hz mark and 2100-Hz space frequencies. Bandwidths of the MF10s are software controllable to accommodate different received data rates and receiver frequency tolerances. A point-to-point wired prototype of the modem has been built on an S-100 perf board. Power may be taken from the S-100 bus or provided by a separate power supply. The prototype has been laboratory tested with excellent "eye diagrams" on speeds up to 600 baud with some eye closing at 1200 baud. Still pending is a design decision whether to combine an optimal minimum-shift keying (msk) demodulator circuitry for 1200-baud operation with this modem or to make it a separate modem. Upon completion pc boards and documentation will be made available to amateurs.

Baudot Radioteletype - Some Background

High-frequency (hf) radioteletype (RTTY) using frequency-shift keying (fsk) began in the U.S. Amateur Radio Service in 1953 when the Federal Communications Commission (FCC) authorized F1 emission in the hf bands. Virtually all teletypewriters at that time were military or commercial surplus. They used a version of a five-unit code (U.S. Military Standard or CCITT International Telegraphic Alphabet No. 2) usually referred to as the Baudot or Murray code. Five-unit teletypewriters used since the 1950s in the U.S. normally operated at a speed of 45.45 baud (60.61 wpm), but a few 56.92-baud (76.68-wpm) and 74.2-baud (100-wpm) machines were also used. Machines available from European sources ran 50 baud (66.67 wpm).

Many fsk demodulators (also called "tuning units" or TUs) were home-brewed by amateurs. However, the design standards were those of military and commercial RTTY demodulators. In the 1950s, the demodulators were normally designed to receive two audio frequencies separated by some multiple of 170 Hz, usually 850 Hz. The favorite frequencies for many years were 2125 (mark) and 2975 Hz (space). In time, amateurs experimented with narrower shifts, particularly 170 Hz, using the audio frequencies 2125 (mark) and 2295 Hz (space), and standardized on it in the late 1970s. Today, virtually all amateur Baudot RTTY demodulators use 170-Hz shift, although many also accommodate 850-Hz and 425-Hz shifts as well. Many hf RTTY stations use transceivers in the ssb mode, sending afsk into the microphone input of the transmitter and obtaining afsk output from the receiver audio stages. Many of these transceivers start rolling their audio off not much above 2000 Hz. As a result, the afsk frequencies 1275 Hz (mark) and 1445 (space) (called the low tones) are also used and are standard in Europe.

The majority of these afsk RTTY demodulators were designed as fm demodulators. In this type of demodulator, the signal is first sent through a bandpass filter to remove out-of-band interference and noise. It is then limited to remove amplitude variations. The signal is fm-demodulated in a discriminator or a phase-locked loop (PLL). The output of the detector is run through a low-pass filter to remove noise at frequencies above the baud rate. The result is fed to circuit which makes the decision between binary 1s and 0s.

ASCII Radioteletype

Effective March 17, 1980 FCC rules authorized the use of the American Standard Code for Information Interchange (ASCII) as defined in American National Standard Institute (ANSI) Standard X3.4-1968.

Straightforward asynchronous serial ASCII has not been very popular on the ham bands since it was legalized. This is due to a variety of reasons. Some RTTYers own mechanical teletypewriters, mostly 45-baud Baudot. They currently are being phased out in favor of electronic digital terminals or computers which can communicate in either Baudot or ASCII. AMTOR is now added to the list of modes possible with "glass TTYs."

Probably the reason why ASCII has not caught on in the hf bands is that some amateurs have experienced poor results trying to operate 300-baud ASCII. The lack of success is largely due to the modem design limitations.

In order to operate existing Baudot demodulators at the higher signaling rates used in ASCII, it is necessary to raise the cut-off frequency of the low-pass filter, and it may be necessary to redesign the bandpass filter and any filters used in the detector. The design can usually be stretched to copy 113 bauds. However, poor results can be expected when trying to modify most 170-Hz, 45-baud demodulators to receive signaling rates above 110 bauds.

Other problems inlcude intersymbol distortion due to multipath propagation. Multipath can be reduced or eliminated by operating near the maximum-usable frequency (muf). These problems are discussed in my paper given at the first packet conference.[RIN81]

Amateur ARQ and FEC hf RTTY Systems

Although hf skywave RTTY is difficult, high-quality error-free operation has been achieved by two robust systems using automatic repeat request (ARQ) and forward error control (FEC). One is AMTOR.[MAR81], [FCC RM-41 22], [MEI82] Another experimental system was designed by Jerome Dijak, W9JD.[DIJ81-83]

Amateur Hf Packet Experiments

Here is a summary of U.S. amateur hf packet experimental contacts:

On February 8, 1982 K1RT in Connecticut and W9LLO in California made a brief connection over 20 meters using Collins KWM-380s, Vancouver TNCs and hf RTTY modems.

On May 31, 1982, K8MMO and W4RI, both in Northern Virginia, carried on a two-hour connection on 13 meters, at 1200 bauds, using ICOM IC-701's, Vancouver TNCs and Bell 202 modems.

On October 16, 1982, a 15-minute connection took place between W3IWI in Maryland and and N5AHD in Texas using Vancouver TNCs and Bell 202 modems.

Design Considerations

Data Rates

For this modem design, we are primarily concerned with signaling speeds which are feasible

for passing through unmodified audio sections of Amateur Radio transceivers, particularly hf-ssb transceivers.

The FCC rules permit up to 300 bauds on frequencies between 3.5 and 21.25 MHz, up to 1200 bauds between 28 and 50 MHz, 19.6 kilobauds between 50 and 220 and 56 kilobauds above 220 MHz.

In addition to the regulatory restriction, the upper signaling rate may be limited on hf skywave due to intersymbol distortion introduced by multipath propagation. This and related subjects were analyzed in a paper I presented at the first packet conference in 1981.[RIN81] For some general rreading on the behavior of the hf skywave medium for data communications see [BRA75].

Speeds up to 1200 bauds should be practical on the hf amateur bands whenever a usable frequency exists near the maximum usable frequency (muf). At present, use of a 1200-baud signaling rate below 28 MHz requires a Special Temporary Authority (STA) from the FCC. I plan to apply for an STA in the near future. I also plan to investigate the feasibility of a permanent rules change to permit up to 1200 bauds on all hf bands, including a portion of the 160-meter band where F1 emission is not now permitted.

At the lower end of the speed range, some consideration was given to including the rate of 37.5 bauds for those infrequent occasions when 75 bauds can't be made to work due to intersymbol distortion. The speeds of 18.75 and 9.375 bauds could have been included but were rejected as being too slow.

We decided to make the speeds 75, 150, 300, 600 and 1200 bauds. These five speeds are selected by means of on-board solid-state switches. The modulator clock output rate is controlled by a 4512 8-channel buffered data selector. Filters in the demodulator are set by six 4051 single 8-channel analog multiplexer/demultiplexers. These seven chips are controlled by three lines from the PAD.

Using five speeds and having an 8-channel switching capability permitted the inclusion of wider bandwidths for the three slower speeds of 75, 150 and 300 bauds. The wider-bandwidth capability is to make allowance for frequency error. This is particularly useful when the receiver's frequency is digitally controlled and/or left unattended. The ICOM IC-720A is subject to a frequency error of ±50 Hz when externally controlled. On the other hand, a no-compromise narrower bandwidth can be selected when the receiver is front-panel controlled, thus settable to within ±5 Hz.

Frequency Shift

The 170-Hz shift in common use for Baudot RTTY could be used for data rates of 75 and 150 bauds with signal-to-noise (S/N) ratios common on amateur hf RTTY. Use of this shift at 300 bauds has been done with some sacrifice of demodulator error performance. It is not suitable for the speeds of 600 and 1200 bauds.

When the narrower shifts (say below 400 Hz) are used, there is a tendency for the mark and space frequencies to fade dependently (together). So, if one fades, the other is likely to fade at the same time. The mark and space frequencies tend to fade more independently when the shift is wider. Independent fading is common at the age-old shifts of 425 and 850 Hz, with more independence obseved for 850 Hz shift. Some commercial RTTY demodulators make use of this so-called in-band frequency diversity and use combining and/or selection techniques to continue copying even if one frequency or the other fades completely.

We have chosen a shift of 600 Hz for two reasons. One is that there is enough frequency separation to permit good in-band frequency diversity action. Also, the frequency of 600 Hz is directly related to the baud rates and permits phase-continuous modulation and synchronous demodulation. At 1200 bauds, a 600-Hz shift is called minimum-shift keying (msk) or sometimes fast frequency-shift (ffsk) to connote that the shift is less than 1 Hz per baud.

Physical Construction

We decided to use an (IEEE 696) S-100 card for the modem. This 5- x 10-inch board was about the size needed for all the chips if a double-sided printed wiring is used. The modem can be plugged into an S-100 computer frame and take power from the bus. Or it can be mounted in its own box with a separate power supply if desired. None of the S-100 data or control lines is used.

Having tasted the fruits of receiving RTTY with a polarization-diversity setup, I plan to build a second PAM board for the second demodulator channel. One demod will be fed by an IC-720A transceiver, the other by an IC-R70 receiver. I also just purchased an IC-7072 interface unit which slaves the two together. The first PAM will have the modulator and a demodulator. The second board is to have an identical demodulator (same pc pattern) and, in the place of the modulator, a diversity selector to process the outputs of the two demodulators.

Input/Output Connections

The (data) I/O connection to the PAD follows EIA RS-232-C rules with the exception that the three data-rate control lines (pins 18, 23 and 25) are presently at TTL levels to reduce the PAD chip count. Insulation-displacement connector (IDC) headers are used on both the PAD and PAM boards rather than the bulkier DB-25. If there is need to route this I/O outside the cabinet, a cable with an IDC plug would connect to a back-panel-mounted DB-25.

The (analog) I/O connection to the radio is to be done with an IDC header. Work remains to be done on the radio side of the modem to ensure compatibility with various amateur hf radios. The goal is to devise an interface scheme that will permit the greatest flexibility.

The interfacing of the transmited analog (TxA) and received analog (RxA) signals is only a matter of adjusting levels and keeping unwanted rf at arm's length. However, the IC-720A and the IC-R70 (and some other ICOM radios) have a 24-pin accessory connector which permits digital remote control of frequency. As some external control circuitry is needed for the ICOMs, one of my future projects will be to design a Receiver Interface Board (RIB) to go between the modem and the radio(s) when ICOMs are used. The RIB will also be built on an S-100 card. As the RIB will be optional, the connector scheme will be designed to work with the RIB or without it.

The I/O design has been deferred until receipt of a Tuscon Amateur Packet Radio (TAPR) beta test model TNC. Although designed as a companion modem for the AMRAD PAD, the goal is to make the PAM usable with the preexisting Vancouver and TAPR TNCs before the I/O design is finalized.

Circuit Description

Fig. 1 shows the shematic of the PAM as of this writing. CMOS logic integrated circuits are used throughout.

Modulator Circuitry

A 2.688-MHz crystal master clock feeds two divider chains. This crystal is not on a stock frequency and was ordered from a crystal manufacturer.

The U2-U6 chain divides by 32, then divides by 7 for mark or 5 for space. U4 divides by 8 and feeds its outputs to two U5 Ex-OR gates to produce a stepwise approximation of a sine wave at either 1500 Hz (mark) or 2100 Hz (space). The U6 active filter removes the steps to produce a nearly sinusoidal output to modulate the transmitter.

The other divider chain produces either X64 or X32 clock for the PAD or TNC respectively for the signaling rates of 75, 150, 300, 600 and 1200 bauds. The U9 binary counter delivers a number of outputs which correspond to X64 clock at each rate. These outputs are selected by U10. A divide-by-2 counter of U4 is used for an X32-clock output.

Demodulator Circuitry

A commonly stocked 2.097152-MHz crystal was used as the master clock to 49.94 times the center frequencies of the two MF10 filters. This stock crystal (and divider chain) just happens to clock the MF10 filters for center frequencies of 1499.8 and 2099.7 Hz.

U109 is the MF10 space filter, U110 the mark filter. Both sections of each MF10 are used to achieve a 4th-order bandpass filter with steep skirts to cope with hf band crowding. Four 4051 8-channel analog multiplexers, U105-U108, select resistance values which set filter bandwidths.

The first halves of U111 and U112 are active low-pass filters for the mark and space tones and provide feedback to the MF10s to reduce tilt. The other halves are full-wave detectors.

The first half of U113 sums the detected outputs and does some low-pass roll off. The main post-detection low-pass filtering is accomplished in the second half of U113 with U114 and U115 there to switch resistor values according to baud rate. U116's two halves are positive and negative peak detectors.

One half of U117 is a comparator for the positive and negative levels and feeds the RxD to the PAD via part of U5 which can be used to invert data. Inverting data is not necessary when NRZI encoding is used but may be needed for NRZ-encoded signals.

The other half of U117 is a tuning indicator circuit. An on-board LED is switched in for testing and out when an external LED is connected. Here's how to use it: While tuning the receiver, control the input level (RxA) for less than half brightness. Tune for maximum brightness, reducing the receiver gain as necessary. Increase the input signal level until a maximum is reached, then back it off to half brightness.

Of course, an oscilloscope would provide a better tuning display. One can be connected at points X and Y fed by the first halves of U111 and U112.

Semiconductors

Q1	2N2222	npn transistor
U1,101	CD4049	hex inverting buffer
U2,9,102	CD4024	7-bit binary counter
U3,7,8,103,104	CD4029	presettable binary/decade up/down counter
U4	CD4520	dual binary counter
U5	CD4077	quad ex-OR
U6	LM741CN	op amp
U10	CD4512	8-channel data selector
U11	MC1489	RS-232-C receiver
U12	MC1488	RS-232-C driver
U105-108,114,115	CD4051	single 8-channel analog multiplexer/demultiplexer
U107,108	MF10BN	universal monolithic dual switched capacitor filter
U111-113	MC1458	op amp
U116,117	TL082	op amp

Acknowledgements

This is to thank Bob Watson for his design work and many patient hours discussing approaches and to acknowledge contributions to the design made by Terry Fox, WB4JFI.

References

[BRA75] Kenneth Brayer, "Data Communications via Fading Channels," New York, NY: IEEE Press, 1975.

[DIJ81-83] Jerome Dijak, W9JD, "The W9JD HF ARQ/FEC System," AMRAD Newsletter series, August, October, November and December 1981, January, February, March, April, November and December 1982, and January 1983.

[EIA RS-232-C] "Interface Between Data Terminal Equipment and Data Communication Equipment Employing Serial Binary Data Interchange," Electronic Instries Association, August 1969.

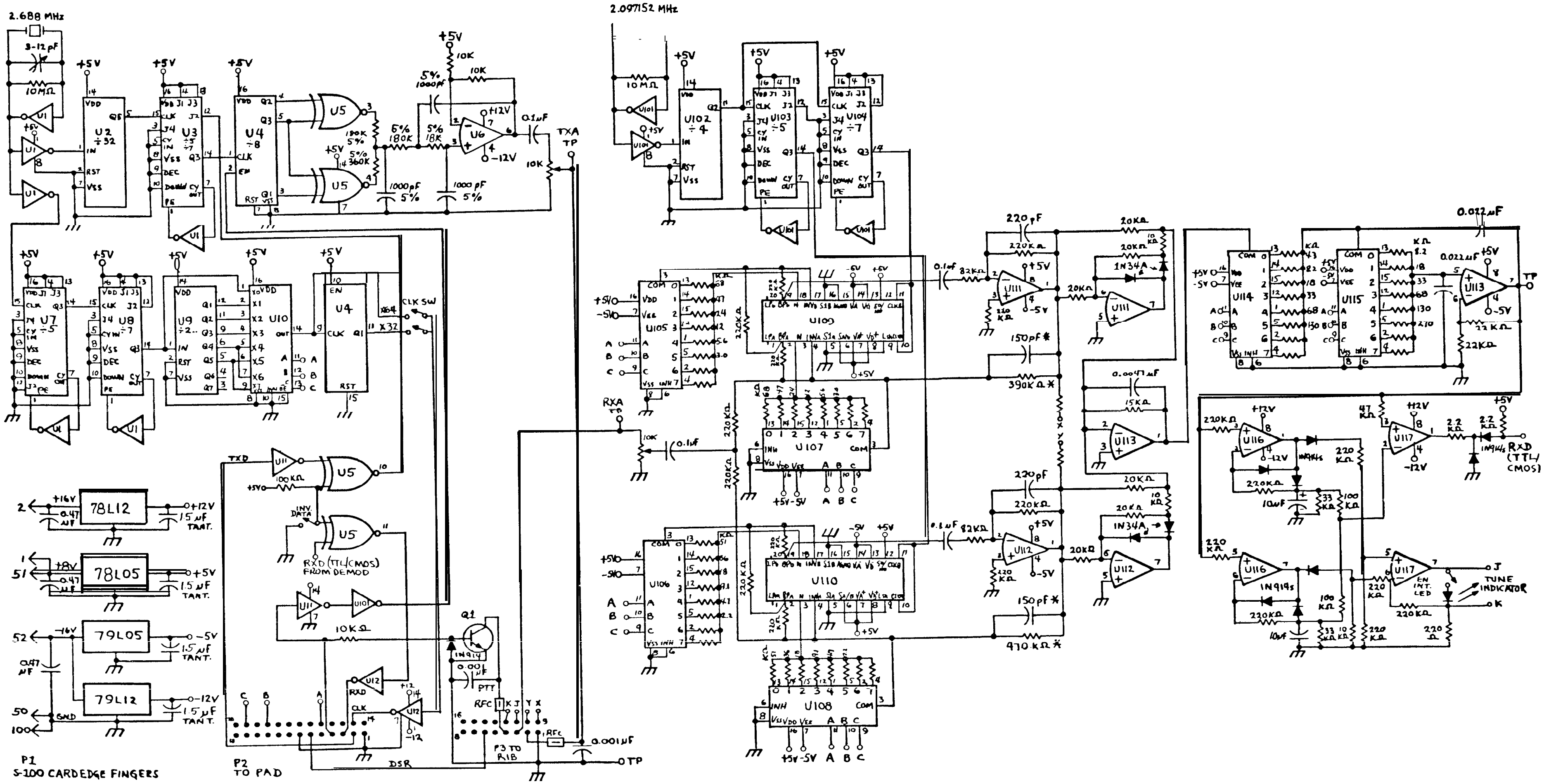
[FCC RM-4122] "Authorization of the digital code "AMTOR" for use by stations in the Amateur Radio Service," PR FCC 83-36 32687, released February 8, 1983.

[MAR81] J. P. Martinez, G3PLX, "Amtor, an Improved Error-Free RTTY System," QST, June 1981.

[MEY82] William Meyn, K4PA, seminar on AMTOR, Teleconference Radio Net, December 2, 1982.

[RIN81] Rinaldo , Paul L. Rinaldo, W4RI, "Amateur Packet Network Agenda," proc. ARRL Amateur Radio Computer Networking Conference, October, 1981.

[WAT80] Robert E. Watson, "FSK: Signals and Demodulation," Watkins-Johnson Company Tech-notes, September/October 1980.


2.688 MHz
2.097152 MHz
78L12
78L05
79L05
79L12
TXA TP
RXA TP
CLK SW
X64
X32
TXD
INV DATA
RXD (TTL/CMOS) FROM DEMOD
PTT
RXD (TTL/CMOS)
TUNE INDICATOR
EN INT. LED
DSR
P1
S-100 CARDEDGE FINGERS
P2
TO PAD
P3 TO
RIB

SOFTNET - AN APPROACH TO HIGH LEVEL PACKET COMMUNICATION

Jens Zander and Robert Forchheimer
Dept. of Electrical Engineering
Linköping University
S-581 83 Linköping, Sweden

Abstract

SOFTNET is a packet-radio concept under development in Sweden. The network is distributed and all nodes are programmable via the network during normal operation. This concept represents an unconventional approach to the protocol issue and offers elegant solutions to the higher level communication problems. This paper gives a programming model of the network, along with some illustrating examples.

I. Introduction

The SOFTNET approach was conceived in 1980 and discussed among Swedish radio amateurs. The discussion led to a proposal for an experimental network in the 432 MHz band utilizing bit rates up to 100 kbps. During 1981 this draft was presented to the Swedish Telecommunication Administration. The Administration responded in a positive way, giving the packet radio group at Linköping University virtually free hands. This group, consisting of 6 people, is currently involved in developing prototype nodes and basic software for the network.

The main concept behind SOFTNET is that all packets are considered to be programs of a network language. These programs are interpreted in the nodes as soon as they arrive. Nodes can be programmed by any number of users simultaneously without unwanted interaction. This approach makes it possible for a user to define his own high level services like datagrams, virtual calls, file transfers and mailboxes. The concept also allows changes at lower levels during operation, permitting redefinition of LINK-level/Access protocols. A detailed description of these ideas can be found in [1],[2],[3],[5].

II. Node model

In a SOFTNET node, an incoming packet that has passed the link level is given to the node computer for interpretation. Here, a standardized set of instructions are available. The kernel of this set is simply a FORTH interpreter to which has been added functions that control the node hardware. Thus, any user may execute his own FORTH program in any of the nodes that he can reach. This way he is able to instruct another node to either deliver the packet to the owner of that node or to retransmit it so that the node merely acts as a repeater. FORTH allows the creation of private directories so the user may also store programs in remote nodes. These programs may either wake up upon the arrival of a packet from the user or upon an internal signal (e.g. the real time clock) produced by the remote node itself. Describing the node thus reduces to describing a programming model. In the FORTH case, this is done by simply listing all the available functions or "words". [4]. Fig. 1 summarizes the packet format from the user's point of view.

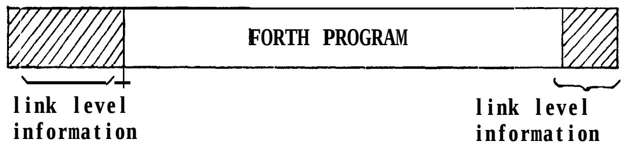


Fig. 1. A SOFTNET packet

In fact, the link level protocol has been added to the FORTH kernel so that also the link level information is handled by a FORTH interpreter. This permits on line reprogramming and extension of the link protocol such as new version of HDLC, access algorithms etc. Thus, from the first byte to the last, a SOFTNET packet is simply a set of FORTH statements. From a practical point of view it is a good idea to conceptually keep instructions at the link level apart from the higher level programs since changes at link level have to be coordinated among the users.

III. The Node - a multiuser/multitasking system

Processing at the link level requires real time performance while higher level tasks are less time constrained. On the other hand, the link processor serves one packet at a time sequentially while higher level tasks may run concurrently. Also, the programming activities of one user should not influence any other. Thus, a SOFTNET node must be able to support parallel tasks besides being able to keep apart the current users of the node. For the prototype implementation our choice was a dual processor (6809) system. One of the processors are solely devoted to link level processing. The second processor contains a multi tasking FORTH interpreter and is shared among the users. A special task - the owner process - interfaces the node to the owner's equipment which can be anything from a dumb terminal to a fullgrown computer system. In the latter case the dual processor FORTH system is simply considered a modem between the owner's system and the network.

IV. Node programming example

Consider the simple network given in fig. 2. Here four nodes are connected by two-way radio paths as indicated by the arcs between the nodes.

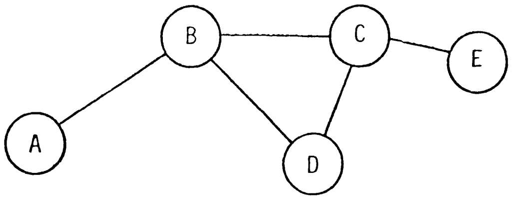


Fig. 2. Sample Network

Suppose a user is located at node A and has the specific task to deliver a large number of packets to node E, i.e. he wants to establish a "virtual" call to E. This can be done in at least two ways. The simplest thing to do is just to add a retransmit command to all packets as is shown in fig. 3a). The command % takes the next symbol as an (one-hop) address and transmits the rest of the packet to that address. This goes on, dropping one address each time, until the remaining packet reaches node E were the data portion is transferred to the OWNER of the node. This procedure may however consume valuable packet space, especially when many intermediate nodes are used. We can instead make use of the programming

```
%B   %C   %E   OWNER <data>                a)
```

```
%B   : VCE ." VCE" %C ;
                                           b)
```

```
%B   %C   : VCE ." OWNER" %E ;

: VCE ." VCE" %B ;

VCE  <data>                                c)
```

Fig. 3. Node programming

feature and instruct the intermediate nodes B and C just to pass along the packet to the next node in line. This can be done as in fig 3b). Here we define a new function, named, say, VCE in the intermediate nodes and our own node. A new definition is made FORTH-style starting with a "• " and ending with a ";". The effect of executing VCE is to pass on the rest of the packet to the next node in line. Also, the function places a copy of its name first in the packet for repeated execution in the succeeding nodes.

V. Project status

Since the advent of the project at Linköping University, a rapidly growing number of interested radio amateurs have joined the discussions. A SOFTNET user group, SUG, is being formed as a subgroup of AMSAT-SM. Up to date this group has received about 1.00 applications for membership.

Hardware development has made considerable progress. The Node-computer board is under production and a first shipment of 50 kits was del ivered in February. Also the PC-layout for the LINK-computer board is completed. The packet-radio utilizes a duobinary direct FSK modulation scheme with favourable bandwidth properties. Transmission is synchronous and MFM coding is used to recover clock information. Due to problems in the design of the radio testing of the digital hardware-and software had to be done on a cable bound local network. A system with up to 4 nodes has so far been successfully demontrated and has provided useful results for further software development.

VI. Conclusions

The SOFTNET concept with its fully programmable nodes will give the user opportunity not only to communicate, but to conduct experiments in network architecture and network protocols. The concept is applicable to all kinds of communication networks. An implementation using a local network cable has been successfully tested and a UHF-radio broadcast network is under construction [2].

VII. References

[1] Persson, I., Forchheimer, R.: Design Considerations of a Distributed Packet Radio Network using the Amateur Radio bands, Internal Report, LiTH-ISY-I-0408, May 1980.

[2] Zander, J., Forchheimer, R.: Preliminary Specifications for a Distributed Packet Radio Network for Computer- and Radio Amateurs, Internal Report, LiTH-ISY-I-0424, January 1980.

[3] Zander, J., Forchheimer, R.: Softnet-Packet radio in Sweden ARRL Amateur Radio Computer Networking, Conf. Gaithesburg, MD, October 1981.

[4] Forchheimer, R., Zander, J.: Softnet - User's Manual, Linköping 1983.

[5] Qvigstad, F., Matts,, S.: Construction of a packet radio Node computer, Internal Report, LiTH-ISY-I-0491, December 1981.

Third ARRL Amateur Radio Computer Networking Conference

April 15, 1984
Trenton, New Jersey

Coordinator:
Paul L. Rinaldo, W4RI

NETWORKING CONSIDERATIONS FOR THE AMATEUR PACKET NETWORK

J. Gordon Beattie Jr., N2DSY
206 North Vivven Street
Bergenfield, NJ 07621
201-943-7754

Abstract

Several issues related to inter-network communication require the discussion and consensus of the amateur community if packet communications facilities are to be flexible for the users and coordinators alike. In this article, the reader will be shown a system of suggested network hierarchies and the interface control procedures required. These hierarchies are patterned after the ARRL's National Traffic System. This is not to say that this writer is suggesting a monolithic structure. In fact, some sections or regions may evolve with several networks covering the same area. This will occur as a result of demand or interest in a particular area. Such a situation is like having early and late sessions of an NTS net, except packet networks can run continuously and simultaneously.

Local Networks

During the past few years, U.S. amateurs have developed local networks using the AX.25 PAD devices. These microprocessor-based controllers convert asynchronous data characters into HDLC frames. The PADS then convert HDLC frames into asynchronous data. This approach allows for a "universal interface" to be developed without regard to the user terminal or computer. It also offloads the frame management functions to an auxiliary processor. The asynchronous link to the user terminal or computer may, at the user's option, support X-ON/X-OFF or RTS/CTS flow control procedures. There are several designs available in kit or semi-kit form. All are supporting the AX.25 protocol and therefore are able to communicate with one another.

Packet Switch Configuration

```
     Section Network Frequency
                 |
            Modem/Radio
                 |
           PAC/NET Board
           (PAD or TNC)
                 |
  Big Board w/Network Database
                 |
           PAC/NET Board
           (PAD or TNC)
                 |
            Modem/Radio
                 |
      Local Network Frequency
```

The AX.25 protocol has only standardized link-level (level two) procedures. "Link level" refers to the control procedures between two stations in direct communication. Some groups have chosen to use the link-level repeater field in order to quickly develop communications beyond the local RF domain. The use of "digipeaters" has helped improve activity in most areas where it has been introduced, but such operation has its drawbacks. Digipeaters operate solely in the local RF domain and they transparently repeat all data with the proper repeater ID. As a result, the operational throughput is cut by more than 50%. Experience has shown digipeaters to be somewhat encumbering when the local network expands. If your station is "lucky" enough to be in close proximity to more than one operating digipeater, you may find the frequency too busy for anything but the most intermittent of transmissions.

Local to Sectional Network Access

The need to communicate beyond the local RF domain will certainly be well served by digipeaters as well as by a more sophisticated network hierarchy. A system of connected local networks or nodes, would remove some of the traffic from the local sub-networks by creating geographically smaller networks and providing links on frequencies other than the local networks. This greatly reduces contention for the local channels and improves local and inter-network throughput.

In order to develop a network consisting of many local networks, procedures will be needed to control

call setup, data transfer and clearing. Special management facilities will be required to route and service the extended network user. The local networks will be provided with these services by special facilities called "Packet switches". The switches will maintain a list of: active stations by establishing a connection to each during idle periods. This list may be accessed to check the availability of another user. If the desired station is down or busy, the caller should be able to leave a "please call me" message for the other user in the switch. This message would be automatically used by the switch to set up the call once the station becomes available. Similarily, if the user is not at his normal location, forwarding information should be left in the switch data base and used to automatically reroute incoming calls.

The switch functions in the local network as any other station, but the switch should have links going on several local and remote frequencies. If a station wishes to communicate beyond the local network, then a connect is made to a local switch port. The switch then presents a menu of options for the user. This would include a call request with routing data from the switch data base, a call request, based on user supplied data, an option to leave a "please call me" message or call forwarding information. Other features, including a mini-message service, could be added if local interest warrants.

Sectional Network

Over the inter-network links many users data can be transmitted. This done by multiplexing the data using an agreeable protocol. The CCITT's recommendation X.25 specifies procedures for this packet level (level three) communication. Some derivation of this would give the amateurs the required procedures to operate the network. There also must be some method of inter-switch communication, This is needed to handle special call request facilities and switch status information. The latter issue is of importance if a cluster of switches is acting as a regional network. If links have been established, the regional network may wish to present itself to the outside world as a homogeneous structure with the means to locate and reroute stations for the calling station.

For example, KA2BQE, located in a local network, which is a part of the Southern New Jersey Section Packet network, wishes to place a call to N2DSY in the Northern New Jersey Section. He establishes a connection to his local switch, then informs the switch of his desire to contact N2DSY in Northern New Jersey. The calling switch has no real idea where N2DSY really is within Northern New Jersey, but he places the call. If the Northern New Jersey section net has enough "smarts" to find N2DSY in one of the switch data bases, then a call will be routed to the switch nearest N2DSY and the link will be brought up. If the link is not brought up or if N2DSY is busy, the call will be cleared with the appropriate cause and diagnostic code. If the Northern New Jersey section network needs more routing information from the caller, the inter-sectional switch in Northern New Jersey will clear the call and may offer a list of local network names to try. This latter information may only be available if the caller specified in a call request facility that such information would be desirable in the event of a call clearing condition. The format and variety of these facilities will require some discussion.

Upper Level Inter-Network Access

The upper level packet switches may have a fairly simple job of routing data, but the volumes that they will be required to handle may be larger than a radio club or even a club federation could finance. No studies have been

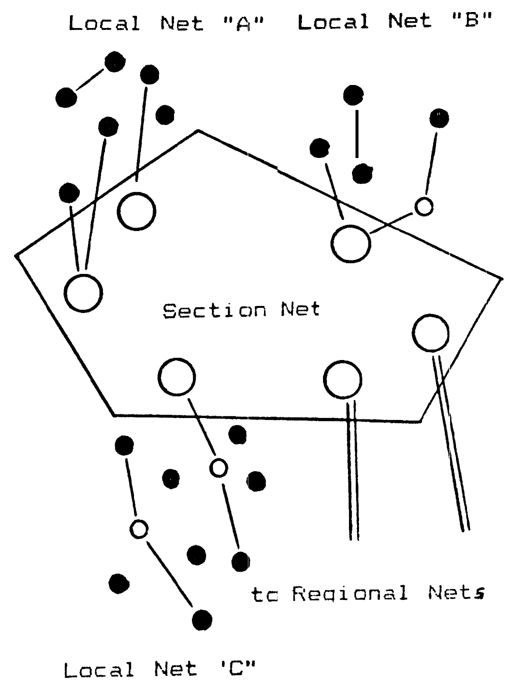
made in this area and it requires our attention. Some surplus commercial equipment could be pressed into service, but it must then be operated, housed, powered and maintained. If real-time facilities are to be available, the amateur community will need significant amounts of money and support for the network.

The variety of sectional network capabilities must be addressed by our packet level procedures. The amateur community may be better off if X.75 single link procedures are incorporated in a single packet level (level three) standard. This would then eliminate the need for a separate standard and implementation. On the other hand, it would require the implementation to be more comprehensive. A call could then be placed via a "transit network". A transit network is one which forwards data as a third party to the call. To properly inform the transit networks of their role, they will require additional control information (inter-network facilities) in the call request. In the event of an unsuccessful call, the clearing station must be able to include the information needed to successfully retry the call.

Summary

The packet switches, initially planned for use in the Northern New Jersey Section by the Radio Amateur Telecommunications Society and the Cherryville Amateur Repeater Club, will consist of Digitgal Research Big Boards and PADS from Bill Ashby and Son. These PADS will perform all link overhead while the CP/M based Big Boards will perform switching and data base functions. The network will have a service area extending from Philadelphia, Pennsylvania to Fairfield County, Connecticut. Other groups interested in extending this are welcome. The network also has a variety of "host" computers available on the network 24 hours a day. A major university has expressed great interest in having their "electronic conferencing" system on the network, and the details are being worked out at this time. A local chapter of the American Red Cross has also started to explore computers and radios in the field to maintain a list of disaster victims and their status.

It should be remembered that this technology will grow rapidly and as it becomes popular, the methods and systems in use will be in a constant state upheaval. We shall see, in the next few years, many changes and improvements - --------- ENJOY THEM !!!



THE EASTNET NETWORK CONTROLLER

David W. Borden, K8MMO
Director, AMRAD
Rt. 2, Box 233B
Sterling, VA 22170

Abstract

This paper describes a proposed packet radio network control computer running at high packet baud rates on the East Coast Amateur Packet Network, EASTNET. Principally discussed is the digital hardware, but also mentioned is some crude RF hardware to accompany the control computer. The digital side uses STD bus hardware developed by Jon Bloom, KE3Z to begin testing, and eventually will use the AMRAD Packet Assembler Disassembler (PAD) board running in an S-100 Bus (IEEE-696) computer.

Introduction

The basis of a real packet radio network is the packet switch, which in its simplest implementation is a two port HDLC I/O board running in a microcomputer.

A Z80 based, STD bus computer has been assembled which is capable of sending and receiving high speed packets, of at least 48K bits/second and possibly 56K bits/second. This computer can form the basis of the network controller, the smart packet switch. It will first be tested on a wire connecting two such devices.

After the packet switch digital hardware is tested, running at some high speed, the required RF hardware must be constructed. Work will begin at 9600 baud, using already proven technology designed by Stan Kazmiruk, VE3JBA. After duplicating Stan's system, the hardware can be upgraded to faster speeds, at least as fast as the digital hardware.

Following the successful testing of the packet switch, the current in-place 1200 baud repeaters can be replaced with the new equipment, thus increasing the thruput.

Background

The essential element of network hardware is the digital packet switch. In the crude kludge, a packet is received containing multiple repeater addresses and the network hardware consists of a repeater which accepts the packet, checks it for correctness (CRC correct), looks for its address somewhere in the repeater fields (up to 8 fields currently allowed), and upon finding its address, the packet is repeated, after setting an agreed upon "has been repeated bit".

A slow network can be constructed with this simple protocol, but it will not meet the design goal of coast to coast packet connections. The real network controller is a true two port syncronous HDLC device that sends and receives packet traffic on both ports simultaneously. More to the point of this paper, it does it quickly (higher baud rate than the local area networks it serves). To achieve our design goal, we require a real IS0 Level 3 protocol with the software for it implemented in our network packet switch. I am leaving the protocol fight to others and intend to provide fast hardware to implement whatever is agreed upon.

Existing Hardware

The Vancouver Digital Communication Group Terminal Node Controller board is currently being used to communicate on the local area networks around the country at 1200 baud. It will not go much faster on the packet side due to the slow clock speed, the noisy 74LS138 I/O decoder and slow 2708 EPROMs. Bill Ashby has been trying to get it to function at 9600 baud and has failed.

The Tucson Amateur Packet Radio (TAPR) terminal node controller may go higher speeds than 1200 baud by not using the on board modem and cranking up the clock speed. However, Tom Clark, W3IWI says the interrupt structure may be overrun at 9600 baud. This represents an unknown at this point. There appears no way to go 48K bit/second or greater speed.

The Bill Ashby terminal node controller has been tested at 9600 baud and works well. It probably will not go faster than that, but Bill is testing it.

The AMRAD Packet Assembler Disassembler (PAD) board, designed by Terry Fox, WB4JFI, exists only as a prototype board currently with plans for making printed circuit boards sometime in the future. It is not available for high speed experimentation, but will be capable of the higher speeds once it is in production.

An STD bus packet radio repeater exists and can possibly be used for high speed work. AMRAD is proceeding along these lines. The repeater is currently working and requires little modification to be used in network service.

STD Blue Box Description

The STD bus was developed commercially by PRO-LOG, and is supported by over 30 manufacturers including MOSTEK. AMRAD managed to obtain some STD computers (referred to as the "Blue Box") at a reasonable price. The computers used the MOSTEK MDX-CPU2 CPU card, a Z80 microprocessor with on board Counter/Timer (CTC) circuit and RAM/ROM sockets for 11K of memory. The heart of the packet work is accomplished with the MDX-SIO Serial I/O module, a serial board based on the Z80 SIO controller. The STD Bus is not the integral part of this system, the SIO is. This hardware could have been placed on the S-100 Bus just as simply. In fact, the designer of this packet project (Jon Bloom, KE3Z) has done just that,, using the California Computer systems CPU card containing an SIO and CTC, however no details of that work are available and the CPU card price is excessive.

The problem with using a Z80 SIO is that it does not do digital phase lock loop (DPLL) recovery of the data derived clock, nor does it do NRZI encoding. Jon solved this problem by building a state machine (prom and latch!) for receive and another for transmit to perform these functions. Details of this design (what is in the proms and schematic) is available from the AMRAD Corresponding Secretary. Using this hardware!, repeater software was written to use this box as a digital repeater. This repeater functions well at 1200 baud while connected to a conventional Bell Standard 202 modem. This box thus forms the basis of the big experiment.

Proposed Experimentation with Digital Hardware

The proposed experinnentation will begin with the construction of a second STD box to hook to the first with a standard RS232 wire cable. The jumpers on the SIO cards are first set at 1200 baud. The current software repeats, but is modular. A short bit of additional software can be easily written to send a packet from Box 1 to Box 2 (running the repeater software). Box 2

will repeat the packet and send it back to Box 1, which will receive it and display it on the system console. Then, the jumpers are moved in both boxes to run successively higher speeds until a packet is no longer returned.

Jon's estimate is that 48 K bits/second is going to work fine, but 56K bits/second may prove a problem.

Another possible upgrade could be made to the STD hardware with a little effort. The SIO board and state machine board could both be replaced with a 8530 Serial Communications Controller (SCC) board. This is the same chip used on the AMRAD PAD board. A kludge board has been purchased for this purpose. This makes good sense as software could be easily developed in the STD box and transferred to the PAD board with few changes. This would be actually easy as very few support chips should be required to put the 8530 on the STD bus. No state machines would be required as the 8530, like the 8273 and 1933, performs the required DPLL and NRZI functions.

Another additional benefit could be derived by keeping the old SIO board around. Two independent channels of HDLC could be run, thus achieving our true network control easily. The only drawback is the required development time as the SIO-State Machine board works now.

Assume for the sake of discussion that 48K bits/second works properly. Then, RF boxes is the next logical step. The jumpers can be returned to 9600 baud and proceed to the next step.

Proposed Experimentation with RE Hardware

Several years ago, Stan Kazmiruk, VE3JBA, published the designs for the Ottawa Digipeater. This hardware repeated 9600 baud packets using a transmitted signal with modulation of +/- 2.5 KHz which proved to be 12 to 14 KHz wide at 6 db down and not over 22 KHz wide at 30 db down. The design centered around transmitter and receiver modules made by VHF Engineering, a company no longer making these modules. Stan mentioned that Hamtronics was making similar modules that should work. AMRAD has a number of these modules and are now testing them for this high speed work. They appear to be capable of 4800 baud as they are shipped and perhaps greater by disconnecting some crystal filter stages. The principal is clear and is receiving attention in New Jersey also (Bill Ashby's gang are going 9600 baud now). Transmit is done by direct FSK of the transmitter, and place the demodulator at the IF (10.7MHz or 455KHz). Faster speeds are certainly possible. Video techniques should get us to 56K bits/second.

Conclusion

The low speed 1200 baud packet work going on all over the country now is fine for local area work. Higher speeds are required for networking. The hardware/software needed for networking need not be copied by a causual observer as the network packet switch is much more complicated than a TAPR board and not as many are required. We should set our sights at going as high a speed as our allowed 100KHz bandwidth will allow. We should use 220MHz for this work as it is relatively unused over the majority of the country and needs to be saved from the commercial land mobile radio service.

HF PACKETS: Modems and Gateways

"CDR Robert E. Bruninga, WB4APR
Electrical Engineering Department
US Naval Academy
Annapolis, MD 21402

With the increasing Packet Radio activity and need for long-haul linking between Local Area Networks (LANs), a number of experiments are being conducted on the viability of packet radio on the HF bands. The simplest approach is direct application of the existing 202 standard modem tones on HF using 300 baud. Although successful links have been demonstrated over long distances under ideal conditions, the performance degrades rapidly under typical interference conditions. The throughput falls to zero in the presence of interfering signals in the wide bandwidth of the Bell 202 standard modem. This paper will introduce two experiments with alternative modems and will briefly describe the experimental HF to VHF gateway currently operational in the Washington DC area,

"The Packet Adaptive Modem"

An effort headed by Paul Rinaldo W4RI is aimed at development of a packet adaptive modem (PAM) which will be able to dynamically adapt to band conditions. The PAMs would use the re-try count as a figure of merit to adjust the baudrate and bandwidth up or down according to a standard algorithm. The PAM will use minimum shift keying (MSK) which minimizes bandwidth while taking advantage of synchronus detection. Two prototype PAM boards have been constructed and will be tested by W4RI and WB4JFI. A special temporary authority from the FCC will be required to operate at baud rates greater than 300 baud. Initial tests will probably use 600 Hz shift at 600 baud.

"Bell 103 Standard on HF"

The approach used at this station is an adaptation of the existing radioteletype (RTTY) and phone line technology using a significantly narrower bandwidth than the 202. A plot of comparitive bandwidths between the Bell 202, 103 and a typical narroband RTTY demodulator are shown in figure 1 which suggest the significant improvement in interference rejection of the narrow-band demodulators. HF RTTY has long been characterized by very narrow filters to optimize reception in the cluttered HF spectrum. Unfortunately most of these HF RTTY modems have a keying rate filter matched to the relatively slow baud rates between 45 and 75 baud which prevents their use at the preferred maximum HF packet rate of 300 baud.

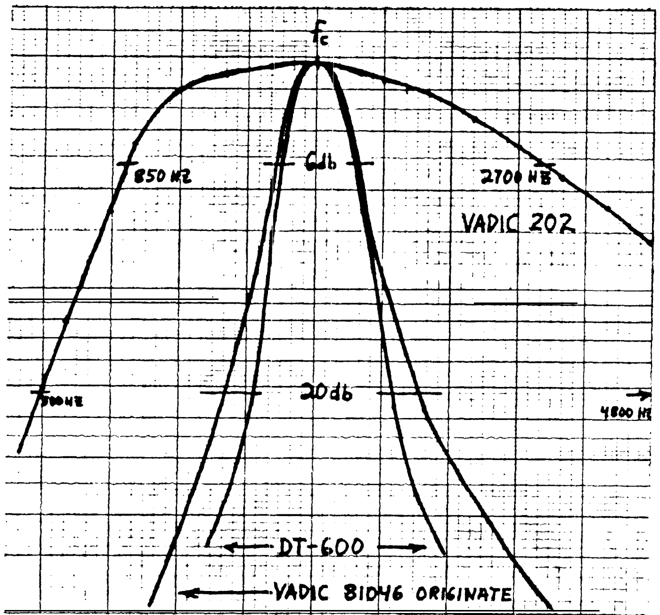


Figure 1. Bandpass characteristics of several modems showing the susceptibility of the Bell 202 modem to interference on an HF channel.

The readily available Bell 103 ORIGINATE modem, however, has a frequency and shift which approximates the narrow bandwidth of a good HF demodulator while having a design speed of 300 baud. The only modification required to use the ORIGINATE modem on packet radio is to modify the transmit tones from 1070/1270 Hz to 2025/2225 Hz to match the receive tones for transceive operation. If receiver incremental tunning (RIT) is available on the transceiver, this modification is not necessarily required. Although the ORIGINATE modem has a bandpass filter matched to the 200 Hz shift comparable1 to the filter used for 170 Hz shift HF RTTF', its performance may not be as good as some of the best RTTY demodulators which use post-detection circuits to correct for selective fading and frequency drift, These corrections are not necessary in the phone line modem and are therefore not included.

Concurrently with my ORIGINATE modem conversion, several stations with TAPIR TNCs were working toward a 200 Hz shift modification f0r the onboard TAPR PLL modem. About mid February, these designs merged and 300 baud packet activity wi.th 200 Hz shift commenced on the new IO Mhz band.

"Present HF Packet Activity"

For the past several weekends, three stations WB4APR in Maryland, W9TD in Chicago, and WORPK in Iowa have regularly exchanged packets under a variety of conditions. At this early stage, each station sends periodic beacon packets to allow frequency netting and propogation determination prior to attempts at establishing a connection. On this band, WB4APR and W9TD have experienced good packet conditions throughout the full test period of 10 AM to 10 PM local time. A large portion of the time, 90 percent of the packets are received without error or re-tries. Presently there is not a problem with interference on the relatively new 10 Mhz band. The frequency used for packet activity is 10.147 Mhz which is slightly above the center of RTTY activity around 10.140 Mhz. The key to successful connectivity is frequency stability and accuracy. For this reason, crystal controlled operation is being contemplated for serious link operation.

"The WB4APR-5 HF Gateway"

As a preliminary demonstration of the capability to extend local area networks (LANs) via HF, an experimental gateway to the Washington DC area VHF packet radio net was constructed as shown in figure 2. Two VADCG terminal node controllers (TNCs) are connected back-to-back through a VIC-20 gateway. The TNCs are available on their respective channels for connects in the usual manner. Upon detecting a connection on either channel, the Gateway program running in the VIC-20 acknowledges the connection with a brief prompt. The user may then select a number of options to assist him in establishing a connection.

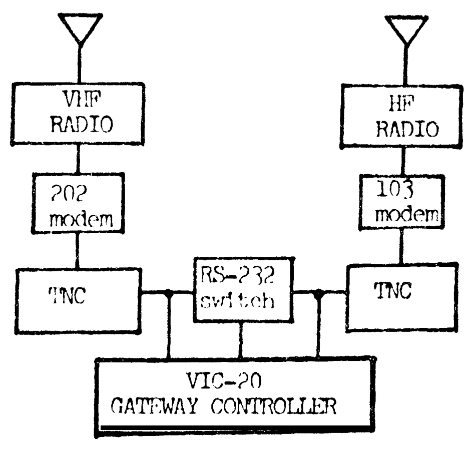


Figure 1. The WB4APR experimental HF gateway into the Washington DC area VHF packet radio net implemented with back-to-back TNC's and a VIC-20.

"Gateway Program"

Although the purpose of the gateway is to provide transparent data connectivity from one channel to the other, in the absence of a Level III protocol, this experimental gateway offers several options as shown in figure 3. In addition to the connect and disconnect commands it offers a list of users available on either channel. In keeping with existing TNCs, it allows the user to monitor packet activity on the other channel in an unconnected mode; a variable beacon capability may be selected to assist in further HF experiments; and lastly, it provides a message capability to the system operator from either channel.

```
** AMRAD HF PACKET GATEWAY **

-- COMMANDS -----------------

HELP (or ?)
USERS HF (or VHF)
LOG (shows log of usage)
INFO (describes system)
BEACON XX (every XX seconds)
MONITOR (unconnected mode)
CONNECT W4XYZ (to sta W4XYZ)
DISCONNECT
GOODBYE (or BYE)
```

Figure 3. Commands available to the gateway user via the VIC-20 gateway controller.

"Message Store-and-Forward"

It becomes obvious that the gateway controller could serve both networks with a store and foreward message capability. Tom Clark W3IWI has suggested that the message store-and-forward node controller is the next logical step in improving the connectivity of local area networks in the absence of a level III protocol. To this end, Terry Fox WB4JFI is assembling a S-100 computer system to run readily available CB'BS software on the local VHF packet net. As the existing gateway software is improved In the future, the HF gateway and CBBS system will hopefully be integrated to take advantage of common hardware.

"Future Gateway Hardware"

The obviously ineffecient back-to-back method of implementing a gateway will be significnatly improved by the Packet Assembler-Disassembler (PAD) currently under development by Terry Fox. This S-100 board will be a complete stand alone packet board which features two Packet serial devices and sufficient RAM and ROM to support the gateway and future level III software. This board is currently in the prototype stage.

EASTNET
AN EAST COAST PACKET RADIO NETWORK

CDR Robert E. Bruninga, WB4APR
Electrical Engineering Department
US Naval Academy
Annapolis, MD 21402

The idea of a digital packet radio network linking the East Coast was envisioned in the late 1970's when the Department of Communications in Canada and later the Federal Communications Commission in the US authorized the transmission of digital data over amateur radio frequencies. Today, EASTNET is a reality with relay sites becoming operational in Washington DC, Maryland, New Jersey, New York, Connecticut, and Boston. By 1985, connectivity from Boston to Norfolk will be established. This paper will discuss the present status of EASTNET and will propose an orderly plan for development of a more sophisticated, higher data rate system. Repeater siting considerations and frequency planning will be addressed.

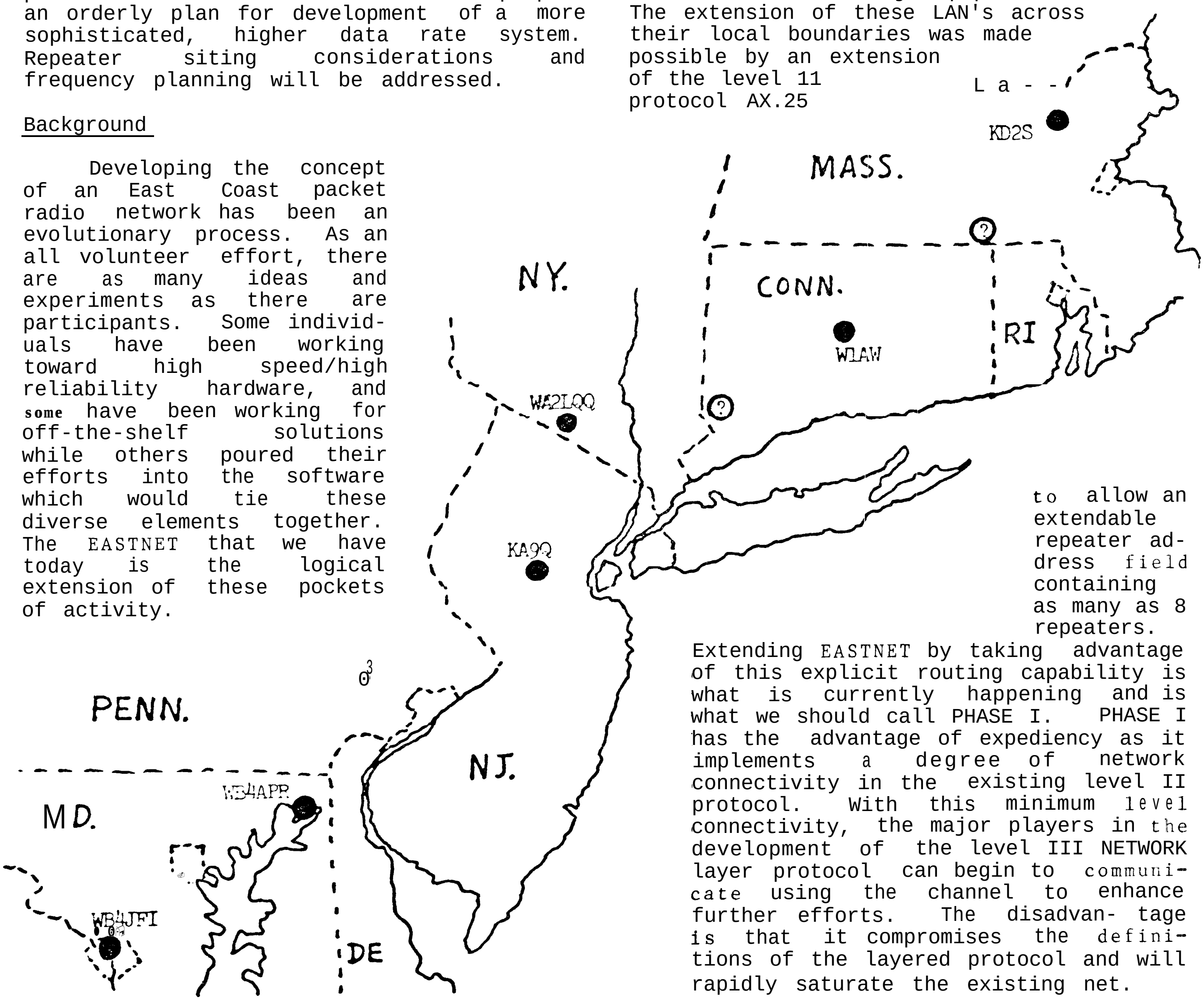
Background

Developing the concept of an East Coast packet radio network has been an evolutionary process. As an all volunteer effort, there are as many ideas and experiments as there are participants. Some individuals have been working toward high speed/high reliability hardware, and some have been working for off-the-shelf solutions while others poured their efforts into the software which would tie these diverse elements together. The EASTNET that we have today is the logical extension of these pockets of activity.

BY proper orientation of these interests, the growth of packet radio can proceed smoothly to faster data rates and more sophisticated network capability while retaining lower level gateways to casual users. This appeal to the entry level user must be maintained in order for packet radio to flourish.

PHASE I

Using the level II protocol AX.25, local area networks (LAN's) have flourished in several of the major population centers. The extension of these LAN's across their local boundaries was made possible by an extension of the level 11 protocol AX.25 to allow an extendable repeater address field containing as many as 8 repeaters.



Extending EASTNET by taking advantage of this explicit routing capability is what is currently happening and is what we should call PHASE I. PHASE I has the advantage of expediency as it implements a degree of network connectivity in the existing level II protocol. With this minimum level connectivity, the major players in the development of the level III NETWORK layer protocol can begin to communicate using the channel to enhance further efforts. The disadvan- tage is that it compromises the definitions of the layered protocol and will rapidly saturate the existing net.

PHASE II

The amount of time that will be required to define and develop a level III NETWORK layer protocol will be significant but will allow a PHASE II improvement in the EASTNET link hardware. The goal under PHASE II will be to move packet radio activity to a significantly higher baudrate in preparation for the eventual implementation of the level III NETWORK protocol under PHASE III. Also under PHASE II, integration of gateways to other channels such as HF and satellite longhaul links will be developed. The hardware will be improved, frequencies established, and permanent repeater sites located.

PHASE III

In PHASE III, the level III NETWORK layer protocol will be implemented within the PHASE II high reliability link hardware, with gateways as necessary to match the topology of the network.

Frequency Considerations

The majority of packet activity is currently centered on two meters because of the overwhelming popularity of this VHF frequency band. The early use of this band for packet radio has had the advantage of attracting new users because of the availability of equipment and ease of integration into existing operation. The explosive growth of packet radio, however, will quickly saturate the available frequencies in this very active band and require new frequencies to be identified.

The 220 Mhz Band

The ideal area for expansion of packet radio is on the 220 Mhz band. This band is relatively under used because historically there has not been as much commercial equipment available as there has been for two meters and 450 Mhz. In fact, the under utilization of 220 Mhz has made it a target for several attempts by commercial interests to obtain the spectrum space from the FCC. Not only is the spectrum space presently available for wideband packet radio expansion, but if the present rate of packet radio expansion continues and a specific band segment on 220 Mhz can be dedicated to packet radio use, the amateur radio community might finally be able to prove conclusively to the FCC the tremendous utility of this band.

Discussions with Paul Rinaldo W4RI at ARRL and the local repeater coordinating committee have suggested the following band plan for packet radio which would be compatible with the existing ARRL plan.

220.55		221.01	
220.65		221.03	
220.75	100 Khz	221.05	20 Khz
220.85		...	
220.95		221.19	

The key elements of this proposal are the allocation of 220.5 to 221 Mhz for experimental use and the overlap of US and Canadian privileges which allow 100 Khz bandwidths in that segment. Five 100 Khz channels would be established, Narrow-band 20 Khz activity could go immediately above 221 Mhz where wideband modulation is not permitted for the Canadians. Since the current allocation of the 221.0 to 222.0 Mhz segment of the band is for link and control frequencies, application for narrow-band packet radio use must be made through the local frequency coordinating bodies. If enough applications are filed for narrow-band 20 Khz packet radio frequencies on the low end immediately above 221.0 Mhz, eventually maybe a 200 Khz segment for ten 20 Khz channels could be standardized. Packet radio groups contemplating experimenting with 9600 baud on 220 Mhz should request coordination of these frequencies not currently in use beginning at 221.0 Mhz and up in their local area. Allocation of the 5 wideband channels should be addressed at a national level, recognizing that hardware will take longer to develop,

Backbone and Gateways

As level III NETWORK layer software becomes available, a single network controller will be defined for each community. This controller will be the Data Communication Equipment (DCE) to all Data Terminal Equipment (DTE) in the Local Area Network (LAN). Any number of links or gateways to other networks, can be logically connected as DTE's or the controller can have transparent connectivity to nearby noc'es or long haul backbores. The details of this connectivity will be addressed in the level III standard, The EASTNET repeater sites during PHASE II will be upgraded to 220 Mhz 9600 baud capability while retaining the two meter gateway frequencies of 145,010 Mhz for local use and access. By the time level III is ready, 56 Kbaud wideband channels will be implemented which can handle the backbone traffic of both the two meter low speed and 220 Mhz high speed local channels.

Repeater Site Selection

The goal of PHASE I is to locate good, permanent link sites with optimum spacing to minimize the number of repeaters required in the backbone system. With thousands of two meter repeaters across the country there is a wealth of experience with the reliable communications range of two meter voice operation, This experience, however, is based on the usual mobile/fixed station-to-repeater path and does not include the more advantageous repeater-to-repeater path to be used in optimum backbone repeater siting. For the Elk Neck repeater site, the Washington DC repeater is barely detectable at ground level using a good mobile antenna, whereas

100 feet up the tower the repeater is well received on a walkie-talkie. The following paragraphs help quantify the expected performance of a repeater site from data obtained using the Washington DC repeater and a portable packet station consisting of an ICOM IC-2AT, VADCG TNC, and VADIC 202 modem.

Path Loss

Figure I shows the typical microwave line of sight plot using an equivalent earth radius factor of 4/3 for the Washington to Elk Neck path. Clearly, an absolute line of site path is not necessary at 145 Mhz for reliable reception.

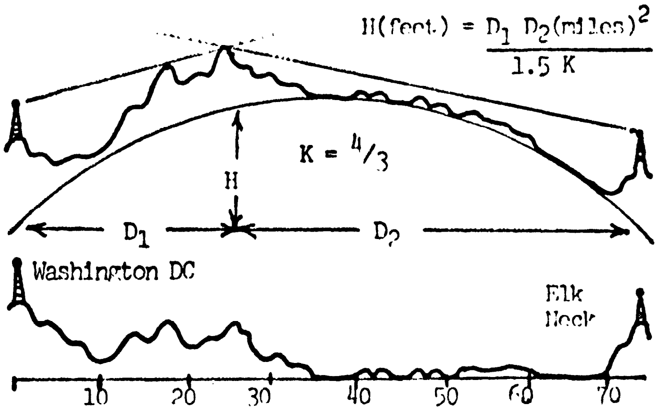


Figure 1. Line-of-site path over flat earth and 4/3 equivalent earth curvature. Total path length of 75 miles.

A plot of tropospheric path loss vs. distance from the ARRL VHF Manual is reproduced as figure 2 and shows a definate flattening of the curve at 200 db from a range of 120 to 200 miles. Although these ranges can be easily operated with weak signal equipment such as CW and SSB, they cannot give the 20db signal-to-noise (S/N) ratio necessary for the relatively wideband AFSK FM modulation techniques currently employed using the typical omni-directional medium gain antenna of a packet repeater.

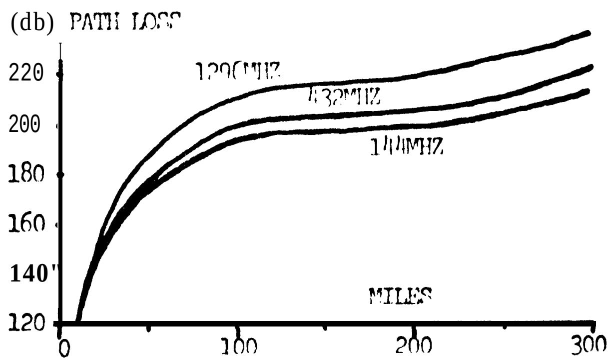


Figure 2. Tropospheric path loss vs. distance for VHF amateur frequencies for 99 percent reliability. (reprinted from the Radio Amateur's VHF Manual)

Receive Margins

Using the portable packet station at several distances from the Washington repeater, packet receive margins were taken with a step attenuator and normalized to a zero db gain receive system consisting of the IC-2AT receiver. These receive margins plotted in figure 3, can be used to predict packet performance at a particular site relative to the 40 watt ERP transmitted signal at an antenna height of 200 feet. To use the plot, simply subtract feedline loss, and add antenna gain and antenna height gain from the nomogram in the VHF Manual. Antenna height gain on paths of about 55 miles is roughly Odb at 30 feet and increases by 4 db for each doubling in height. Although these measurements are only accurate to 3 db or so, they do show that repeater separations on the order of 50 to 60 miles are achievable with nominal sites and equipment.

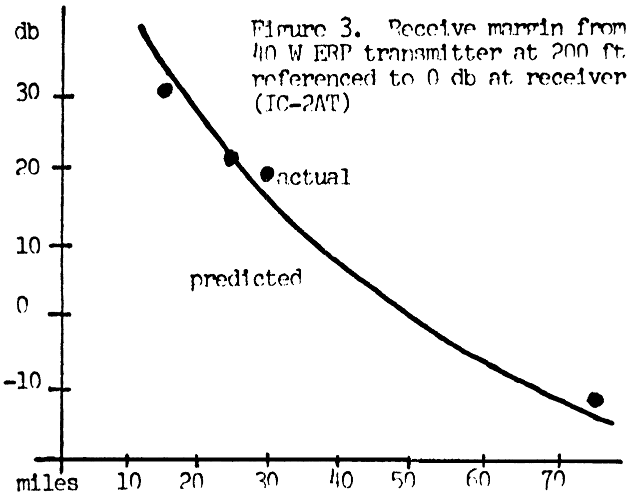


Figure 3. Receive margin from 40 W ERP transmitter at 200 ft. referenced to 0 db at receiver (IC-2AT)

The plot can be used for any other transmitter height and ERP by appropriate corrections. The reliable path between two 40 Watt ERP repeaters at 100 feet with 6 db gain antennas, 2 db line loss, and 7 db receive margins should be about 50 miles. The same stations at 60 feet should be no further than 40 miles apart. Mountain top sites have an additional advantage which can add as much as 35 miles per 1000 feet of altitude over average intermediate terrain. A direct comparison of this data to the calculations illustrated in the VHF Manual suggests using 1 db for the noise figure, 12 Khz bandwidth for the receiver, and a signal to noise requirement of 11 db. Because of the FM improvement factor, an 11 db carrier to noise ratio results in approximately an 18 db signal to noise ratio for the 202 modem. Using these assumptions, calculated path margins using the VHF Manual procedures support the empirical data.

EASTNET Site Summaries

Each of the following summaries is the best information available at the time of this writing (mid March 84). The order is roughly south to north.

Hampton, Virginia

Bill Holmes W4UMC has operated his repeater in the Hampton Virginia area for over a year, Being 140 miles from the Washington DC repeater, he will probably be linked via Richmond and Fredricksburg.

Waynesboro, Virginia

Fred KF4DQ speaks for a group of operators interested in linking down the Shenandoah Valley from the Washington DC repeater. They have the Blue Ridge mountains to cross, but may be able to use that to advantage as altitudes of 3000 feet might be possible.

Washington DC

The AMRAD WB4JFI-5 repeater has been in operation since 1982 on the AMRAD voice repeater but was moved to its present site on the channel 9 TV tower in October 1983 replacing the WB5MMB simplex repeater in Vienna, The tower is on the highest ground in the District at 400 feet but suffers from a very high RF environment. It was moved to 145.010 Mhz in January 1984 coincident with the decision to burn-in that frequency for packet use on the east coast. It uses a VADCG board and runs 15 watts through a 10 inch cavity into an unknown abandoned VHF high band antenna about 200 feet up the tower.

Elk Neck, Maryland

The WB4APR-6 repeater has been in operation since February 84 but was moved to its present site on a 120 foot state forrestry tower at 285 foot ground elevation on 8 March 84. The site has 40 miles of its 75 mile path to Washington over the Chesapeake Bay. It has good visibility into the Wilmington and possibly Philadelphia area with a slight shadow around Iron Mountain. The repeater is a good beacon, but suffers from poor receiver sensitivity and an undetermined transmit tone anomoly which degrades long packets. Replacement hardware is under construction which should allow connectivity with the Washington repeater.

Philadelphia, Pennsylvania

No optimum site has been identified although the TelCo. Pioneer Amateur Radio Club has expressed interest and has a downtown building top location. Steve Robinson W2FPY reports a possible site on the 125 foot Burlington municipal tower on a ground level of 85 feet; but this area is about 15 miles further away from Elk Neck and signal strength is low.

Warren Township, New Jersey

Phil Karnes KA9Q has been beaconing on 145.01 Mhz since February using his TAPR TNC. He is located on a very good hill with excellent southern visibility. He has possible access to the top of the hill (500 feet) and a tower which would improve his shot to the north. He reports good connectivity with W2LQQ in Warwick NY. A possible conflict is RFI to his satellite activities.

Warwick New York

Rip W2LQQ has also been beaconing on 145.01 Mhz using his TAPR TNC and reports favorable connectivity down to KA9Q. He is presently using his OSCAR array and would also have to solve the RFI problem with his satellite activity.

Stamford Connecticut

Ed Kalin K1IRT is interested in joining EASTNET and possibly serving as the link site towards Newington. His visibiliy over to Long Island is also favorable. Path lengths to Warwick and Newington are reasonable if a suitable high altitude site can be located.

Newington, Connecticut

As the national headquarters of the ARRL, the W1AW packet repeater is a key site in EASTNET. The repeater has been on the air since January 1984 using a VADCG board at the ARRL station. This site is not optimum and will hopefully be moved to a higher location possibly at 1000 feet to enhance EASTNET connectivity. Paul Rinaldo W4RI is the point of contact at the ARRL.

Rhode Island

A site located in northwest Rhode Island or south central Massachusetts would have reasonable 50 mile paths to link Newington to Boston. No activity reported.

Boston Massachusetts

The present packet repeater in the Boston area is KD2S in Lowell about 28 miles northwest of downtown. This repeater coupled with a good site southwest of the city would help in extending EASTNET down toward Newington. Efforts to obtain a good mountain top location could extend coverage throughout the state.

EASTNET Coordination NET

To facilitate inter-site communication for the further development of EASTNET, an AMRAD Packet Radio Users Group Net will be maintained at 2200 EST on 3855 Khz every Tuesday evening immediately after the regular AMSAT net. Participants in EASTNET are encouraged to check in for the regular dissimination of EASTNET and general packet radio information.

The Racing Problem: A Packet Solution

Robert E. Bruninga
6103 Hillmeade Rd.
Bowie, Md. 20715

Background

Last June several members of AMRAD participated in the Old Dominion horse ride and 100 mile endurance run near Front Royal Virginia by providing mobile and emergency communications. A dozen or so checkpoints were manned by radio amateurs as well as shotgun riders with each of the key event and emergency personnel to provide VHF communication throughout the several county area of rural roads and 1500 foot mountains. A portable repeater was constructed out of two Icom 2AT walkie-talkies and battery powered throughout the weekend event. This was only the beginning of the excitement involved in designing a better way to do it next year, In fact, our primary interest as noted in Dave Borden's AMRAD Newsletter Packet column of July 83 was the desire to link a system of computers on packet radio to handle the data on the over one hundred horses, riders and runners so that information would be readily available at key points for emergency purposes. This paper will describe a distributed data base system implemented with Commodore 64 and VIC-20 Computers linked via amateur packet radio.

Data Reporting

Knowing where all the horses, runners and emergency personnel are located is the single key to rapid response in emergencies. Keeping the time of arrival and status of all race entries at each checkpoint in a large array space would allow easy manipulation of the data to serve a variety of needs. Since series of numbers are already assigned to the 100 and 50 mile horses and runners, a series of numbers assigned to key personnel and emergency services would allow them to be tracked in the same data system throughout the course. A single reporting format can be used for any horse, person, or asset of the form as shown in figure 4.

NUMBER, TIME, LOCATION, DATA

These reports may be used directly to update an N-by-L array of time and status for every unit in the event. Here, N is the number of persons and horses and L is the number of reporting locations. Actually two arrays with the first one recording the time of arrival and the second one containing a coded status or departure time would hold the entire status and historical file of the event.

Data Fields

BY coding the checkpoint names and status, a very short fixed length string may be used for transmitting this data across the data channel. With a three digit number, three digit time, single character checkpoint and status key, this string is only 8 characters long. The format of figure 1 has been designed not only to be as short as possible but also to require a minimum of processing overhead.

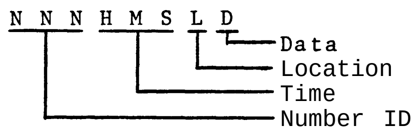


Figure 1. The format of a single race item report consisting of 8 ASCII characters.

The generality of this report is accomplished by assigning numbers to everyone and everything in several compatible number series. For the Old Dominion Ride and Endurance run two such number series have already been used. They are the 100 series and 500 series for the one hundred and fifty mile entries respectively. Using these numbers as a starting point and making some judgments on potential group sizes, the numbering scheme shown in figure 2 is suggested.

l-99 - Ride, Radio and emerg. pers
100's - One hundred mile entries
200's - Runners
300's - Runners
400's - Cavalry
500's - Fifty mile entries

Figure 2. Number series to uniquely identify all race entries and race assets.

Associated with each of these items at any one time is a location. The locations might be gates or checkpoints, or they might take on other meanings as well. Since the data packet going over the radio channel needs to be as short as possible, these location identifiers should be keyed into a single character format. This allows up to 91 keys using the ASC and CHR$ functions over the range 35 to 126. in this way, the time is also compressed into three ASCII characters, one each for hours, minutes and seconds. Some keys for locations along the trail route and elsewhere are shown in figure 3.

ASC	CHECKPOINT	ASC	LOCATION
35 -	4-H camp	55 -	Broadcast msg
36 -	Lands Run	56 -	Front Royal
37 -	Bentonville Br	57 -	Luray
38 -	McCoys Ford	58 -	Detrick
39 -	613 split	59 -	King Crossing
40 -	Yates	60 -	Pala's
41 -	Bixlers Br	61 -	Repeaters
42 -	Woodstock Gap	62 -	Mobile
43 -	50 Finish	63 -	Food
44 -	Edenton Gap	64 -	Gas
45 -	Hickory Lane	65 -	Searching
46 -	Virginias	66 -	Vets
47 -	Seamens	67 -	Unavailable
48 -	Picket Springs	68 -	Gone home
49 -	Shermans Gap		
50 -	613 Split		
51 -	McCoys Ford		
52 -	Lands Run		
53 -	100 Finish		

Figure 3. A table of suggested location identifiers by their ASCII equivalent codes.

The single character keyfields allow some 91 different locations and status indicators to be represented. Several suggested data keys which can serve horses, runners and VIP's alike are as follows:

I - IN	O - OUT
H - HEADED FOR	L - LOST
M - MESSAGE FOR	V - VET NEEDED
S - SCRATCHED	P - PULLED
C - CREW NEEDED	D - DOCTOR NEEDED
F - FURRIER NEEDED	E - OUT TO LUNCH
ETC	

Packet Radio Data Distribution

Recognizing that the most important real time data on any item in the data base is only the last reported status, the organization of the data network as a distributed system could make it much more survivable in an amateur/portable environment. Also the KISS principle (Keep it Simple, Stupid) could be followed more closely. Under this concept, there is no central computer and all the field display computers need to maintain only the last known status on any single item in the data base. To minimize data channel load yet provide for full refresh of the data at the display terminals, a scheduler in each computer would periodically retransmit all data entries for which it was responsible. Because of the one-to-many distribution of the updates to all other display terminals, the packets will be transmitted and received in the packet monitor mode error free, but without acknowledgment. Sending one of these refresh packets every number of seconds would assure complete data base refresh every few minutes or so. With 8 characters per data item, a concatonated combination of eight such items per line would make a nice size packet as shown in figure 4.

```
#uTNNNHMSLDNNNHMSLD......NNNHMSLD
   item #1 item #2       item #8
  Transmitting station identifier
 Update identifier
```

Figure 4. A single packet frame consisting of eight individual item reports.

This single string format allows simple Basic input commands to be used to input the complete string at the display terminals in one operation and minimize packet overhead. The line can be verified prior to array update by testing for the update identifier and fixed length of 67 characters. If a display terminal crashed or lost its data, it could be allowed to slowly rebuild its data or it could be completely updated by any other computer in less than 30 seconds on a dedicated basis at the 1200 baud channel capacity.

Display Processing

Once received, the updates are entered into a string array of the form S$(N)=HMSLD where N is the item number (which may have to be hashed for effecient use of array space if multiple item number series are used) and HMS, L and D are time, location and data keys respectively. The individual display terminals then can be programmed in a variety of ways to provide querie/response information on the latest status. Several possible screen display formats such as the following could be implemented:

* locations of VIP's and Emergency pers
* Top 20 Horses
* Last 20 Horses
* Horses at location X
* Status of Horse Y
* Horses departed location X
* Missing horses
* ETC

The possibilities are endless and free to the imagination of the display terminal programmer. This flexibility is necessary due to the variety of computers and screen size formats which will be used.

Notice that nothing prevents a single computer from building the complete set of N-by-L arrays since all of the data will have been transmitted on the channel. In fact, 16K or larger and Disk based systems should be programmed with this capability. Also note that as checkpoints are completed and all race entries have passed through a particular point, there is no further requirement for packet refresh from that station. That packet station can then be moved to a later checkpoint for reuse.

Data Channel Activity

As the flow of the race progresses, data channel activity will tend to migrate from station to station as shown in figure 5. For a race of 160 entries, the starting point station will initially be responsible for all 160 reports, which, in groups of 8 corresponds to 20 packets. If one packet is transmitted every 12 seconds, a complete update is provided every 4 minutes. This cycle will continue until the lead horse arrives at gate two. As reports are initiated from that station, station one at the starting point will see the reported location field change and stop refreshing that entry on a one-for-one basis with reports it hears from the new reporting station. This is very simple since each station uses the location field to identify its own reports which it is responsible for updating. In this manner it can be seen that there will always be a nominal 20 packets per period being transmitted and that the peak load will move as a bell curve distribution from station to station as the race progresses. There will also be a nominal background level of packets from other stations reporting the movement of VIP'S and other status throughout the course.

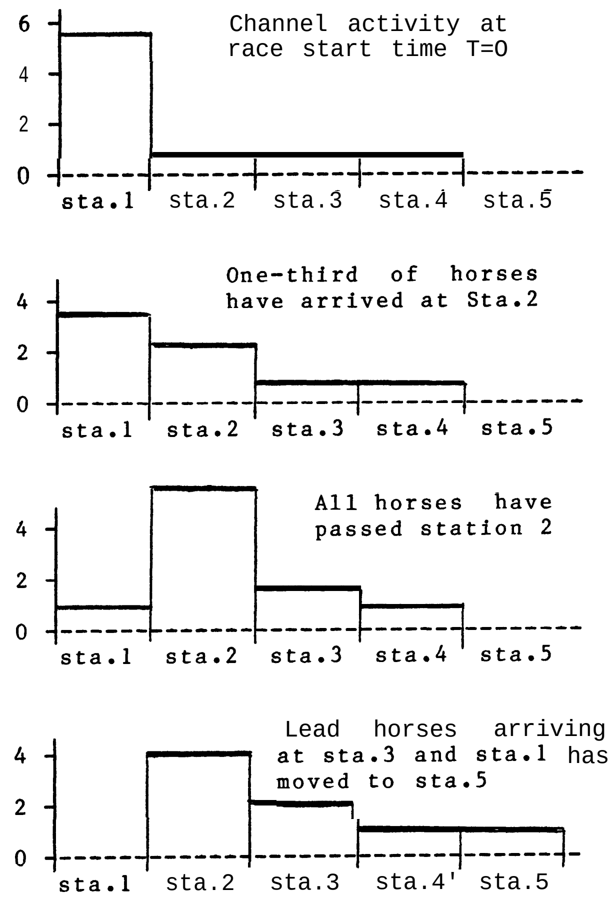


Figure 5. Channel activity in packets per minute remains relatively constant although reporting responsibility moves from station to station.

Station-to-Station Messages

An enhancement to the basic database system described above is the addition of a message packet format that allows the exchange of text messages among the packet stations. The format of figure 6 is used.

```
#mLDNNLLnow is the time for all good men
||||||
|||||└Message text
||||└─2 digit line number
|||└──2 digit message number
||└───Destination (location)
|└────Originator (location)
└─────Message format identifier
```

Figure 6. Message packet format includes line number and message number to assure message integrity over the unconnected net.

Point to point messages are programmed to require a specific acknowledgement, while system-wide messages or broadcasts are scheduled for periodic update. Smaller systems (4K) may retain only the last message while larger systems may desire to retain all messages until cancelled.

Data Array Compression

For array processing, the individual display terminals should be programmed to accept each report and use the time, location and status to update its data base. To save memory, the gaps in the item number series suggest the computation of offset values for each of the number series and using these offsets to compute an actual array address. This means that on initialization, the program queries the user for the total number of entries expected in each of the number groups and uses these as offsets for the array subscripts. Using this approach, the total array size needs to be dimensioned no larger than the sum of these offsets and only a single computation needs to be performed for each array access.

The scheme to significantly compress the number series into a contiguous array is as follows:

```
INPUT"Number of series ?"; NS
FOR I=1 TO NS:
  INPUT"Start and end values?'; S(i),E(i)
  R(i)=E(i)-S(i)        :RANGE
  A(i)=A(i-1)+R(i-1)    :ARRAY OFFSET
  D(i)=A(i)-S(i)        :DELTA OFFSET
NEXT
```

Now, given a horse number H, its array location, A, may-be found using the loop:

```
FOR i=1 TO NS
IF H<E(i) THEN A=H-D(i): i=NS
NEXT i
```

And any array number, A, may be converted back to an item or horse number using the loop:

```
FOR i=NS TO 1 STEP -1
IF A>A(i) THEN H=A+D(i): i=1
NEXT i
```

Finally, some consideration should be given to the scheduling of packet transmissions according to the loading of each particular station. Using 4 minutes as a nominal refresh cycle period, each station should time the delay between each packet transmission inversely proportional to the number of packets it needs to send or a minimum of once per minute if he only has one packet. With a peak load of 200 reports or 25 packets per period, the minimum delay between packets should be about 10 seconds resulting in the following relation:

$$N = INT(R/8)+1 \qquad D = 290/(N+4)$$

Where D is the delay in seconds between each packet transmission, N is the number of packets due for transmission computed from the number of reports R.

The purpose of this paper is to suggest early agreement on the format of the data distribution packets so that AMRAD packet owners can begin working on the display formats and querie/response capabilities of their individual systems. They may add as many bells and whistles as they feel necessary, such as pointers to string arrays containing the full names and statistics of all horses, runners and people. Dependjing on these bells and whistles, there should be no problem fitting up to 200 horses, runners and VIP's into less than 4K for a VIC-20.

The requirement to provide emergency communications coverage for large public race events is a frequently repeated amateur radio public service event. Hopefully bringing the benefits of packet radio to bear on this application will demonstrate the tremendous potential for this state-of-the-art mode of communications.

ISO REFERENCE MODEL REVIEW

Terry Fox, WB4JFI
President, AMRAD
1819 Anderson Rd.
Falls Church, VA 22043

Introduction

This paper will review the status and positions of the various layers in the ISO (International Standards Organization) Open Systems Interface Reference Model (OSI-RM) as applied to amateur packet radio.

Some amateurs believe that the OSI-RM provides a good basis for the development of computer networking via amateur radio because of the flexibility it allows. Others see that same flexibility as a lot of unnecessary overhead that takes its toll in reduced throughput and added complexity at actual network implementation. Even the most die-hard supporter of OSI-RM must admit that it is less than optimum, especially at the network layer. I believe however, that it is the best game in town at this point, and what we amateurs have implemented so far falls neatly into the OSI-RM architecture.

Overview

The OSI Reference Model for a modern data communications system is broken into seven distinct levels. The terms level and layer are used almost synonymously whenever the OSI-RM or its levels are discussed. Actually, when describing or referring to the function, level is generally considered the correct term, and when calling a particular level by name, layer is more often used. Thus, the first level of the Reference Model, Level 1 is called the Physical Layer. A small point I admit, but one we should keep in mind.

The seven levels that make up the OSI-RM are:

Level 7. Application Layer (highest level)

Level 6. Presentation Layer

Level 5. Session Layer

Level 4. Transport Layer

Level 3. Network Layer

Level 2. Link Layer

Level 1. Physical Layer (lowest level)

Each one of these levels has certain responsibilities to make sure data travels from a source device to a destination device accurately and promptly.

Each of these levels communicates with its peers along the overall network as necessary, using its associated lower level as the communication medium (except for Level 1, which has no lower level). All information received from an upper level by a lower level should be considered as data and not altered beyond what may be done to enhance communication of the data within that level (this includes any headers required by the upper levels).

It should be noted that there is potential in the OSI-RM for a lot of duplicity of functions, depending on what protocol is implemented at each level, and how complex the resulting network is. This is especially true when the affect of multiple levels of multiplexing data paths is considered, as most levels allow. Simpler network systems may leave out certain levels because they just don't apply, or add unnecessarily to the complexity of the overall system. I would recommend that if any level is bypassed, at least one null character is inserted where that level would otherwise go, leaving the network designer with an "out" in case that level is deemed necessary at a future date.

One of the major advantages of the OSI Reference Model is that it will (hopefully) allow substitution at any one of the individual levels, without seriously affecting the other levels of the overall network. This means that one area can use the same Network Layer, for example, as another area, while implementing a totally different Link Layer protocol. This not only allows for creative implementations at any of the levels, but also allows for each level to suit the need of its medium.

A good example of this might be the creation of different Link Layer protocols depending on the communications medium used (meteor scatter likes smaller frame sizes than VHF/UHF terrestial channels), while using the same Network and higher layers.

As mentioned above, this design does have its weaknesses. Sometimes, the levels need to be broken down further than they are (such as the Network Layer into the Network Sublayer and Internetwork Sublayer), while other times there seems to be a problem effectively separating different levels (the Datagram type Internetwork Sublayer relies on the Datagram Transport Layer heavily for proper operation). This paper will discuss the various levels independantly, and try to account for any interdependance as necessary, starting with the lowest level, and working upward. I will also mention some of the alternatives at various levels, and make some recommendations based on my opinions as of the date of this paper.

Level 1, The Physical Layer

Level 1 is the lowest level in the OSI Reference Model. It is concerned primarily with the "real world" part of sending and receiving data. This is not as small a task as initially thought. There are several parts that make up the whole Physical Layer, including:

Voltage levels.

Data and handshaking signals.

Speed of data transmission and reception.

Order of bit transmission and reception.

Modulator/Demodulator (Modem) types.

RF signalling channels.

All of these different parts have to match each other at both ends before any data can be transferred from one location to another.

Typically, data at the Physical Layer is sent over a radio channel in a serial bit stream. The interface between the users terminal or computer is generally also serial, usually asynchronous ASCII, at speeds between 300 and 9600 baud. In serial operation, RS-232C is the common interface for defining voltage levels, data and handshaking signals, the types of connectors used, and their pinouts.

The data speed and modem type are related to the RF signalling channel used in amateur packet

radio communications. It is very difficult to design a modem that will allow data transfer at a rate of 56kbps (kilo-bits-per-second) over a data path using the HF frequencies. It is beyond this paper to specify optimal data rates and modem types for all aspects of amateur packet radio, rather, I will discuss some of the more common systems presently being used or being actively discussed.

VHF/UHF Operation

There is only one commonly used standard on VHF/UHF at the moment. It is the Bell 202 modem, running at 1200 bps. This is an extremely popular standard in that it affords a relatively fast speed of operation (compared to 60 wpm Baudot), yet does not require special radios or other difficult to obtain equipment. There are a lot of surplus 202 type modems available, along with several simple modem designs. There are even single-chip modems becoming available (such as the AMD 7910) that do the whole modem magic in one IC.

Even satellite operation is being experimented with, using the above mentioned 202 standard. Some users are finding that some modem designs (sucn as the phase-locked-loop modems) are not functioning as well as others, primarily due to the inferior signal-to-noise ratio SSB over a satellite gives as opposed to VHF FM.

There is some experimentation going on with higher speeds, particularly on the 220 MHz band, where we are allowed to run up to 56 kbps. The present experimentation generally consists of speeds up to 9600 bps (the speed where most HDLC chips internal clock recovery circuits start to die), using different modulation and demodulation techniques. One of these is to not use the classic concept of a modulator and demodulator, but rather shift the RF carrier some specified amount at the transmitting end, and take the output of the discriminator output directly from the receiver, before any audio processing. This technique is actually quite old (relatively speaking), some of the early packet experiments in Canada used this technique quite well at speeds up to 4800 bps. The drawback to this system is that it requires the modification of the radios to be used, and may not give enough of an increase in speed to warrant a long-term commitment of time and materials necessary to develop the system.

There is the potential for a lot more experimentation in the VHF/UHF area, including extremely high speeds using microwave RF technology such as gunnplexers.

Meteor Scatter

Some experiments using meteor scatter are in the design stage. These tests will probably be conducted on 6 meters, with stations about 600 to 900 miles apart (optimum meteor scatter range). Due to the extremely short duration of meteor scatter paths, high speeds and small packet sizes will be the order of the day. This may cause special protocols to be developed to reduce the amount of overhead required, and take into account the sporadic nature of this RF medium.

HF Operation

There is some HF packet operation going on now, with the promise of a lot more in the near future. HF can allow a major jump of physical space in a single hop, if the correct frequency of operation is chosen. HF does have its own set of peculiarities to deal with, such as narrowness of the channel bandwith, selective fading of different frequencies within the channel, and intersymbol distortion due to the RF energy taking multiple paths to reach the other end.

Some of the initial tests were conducted on 40 meters using the VHF standard 202 type modem, running at 300, 150, and even 75 bps. The reason for this initial choice was that the equipment was already hooked up and operating, but it was found that this system leaved a lot to be desired. The major problem in this system was the wide bandwidth necessary to be clear of interference (202 modems use FSK with one tone being 1200 Hz and the other being 2200 Hz, resulting in a shift of 1000 Hz, requiring almost the whole 1000 Hz to be devoid of other signals, no small feat on 40 meters).

One answer to this modem problem is the Packet Adaptive Modem designed by Paul Rinaldo, W4RI. and Robert Watson. This modem uses a relatively new technique to amateurs, Minimum Shift Keying or MSK, for the transmission of data. It will eventually be able to run up to 1200 bps with a channel bandwidth equivalent to a 600 Hz shift FSK modem. The design is completed, and some of the boards are being tested now. The finished system will be written up in an upcoming issue of QST.

Another set of experiments being conducted uses a 200 Hz s'hift FSK modem running at 300 bps. Bob Bruninga, WB4APR is among the group testing this system on a regular basis on the 10 MHz band, using surplus Bell 1.03 type modems. The 30 meter band has some real. advantages to the packet user, the main one being the lack of QRM. Bob routinely maintains connections for up to several hours at a time now, implying this may be a reliable method of transferring packets over a medium distance.

The Physical Layer is the only level that maintains an actual "physical" or ''electrical" connection with its peers. The rest of the levels communicate with their respective peers through assigned "logical" or 'virtual' connections. Since these logical connections aren't part of the real, physical world but rather system concepts implemented in computer programs, there must be an actual computer device used to implement these protocols. These computer programs run either as a portion of a mainframe program, or, more frequently, in a smaller, dedicated computer.

Level 2, The Link Layer

All this leads us to the Link Layer. This level is responsible for receiving and sending data from the higher level protocols and sending that data to or receiving the data from the the Physical Layer, respectively. Part of this responsibility is to make sure that data integrety is maintained through the physical devices implemented, and recovering from any errors occuring in the physical world.,

Figure 1 shows several types of devices interconnected as a portion of an amateur packet radio network. Note that there is a separate link layer that corresponds to each Physical Layer.

In order to insure data integrity over the Physical Layer, the Link Layer does several things to the data it receives from the higher levels. Most Link Layer protocols start by taking the data received from the higher level and creating small groups of data, called frames, then sending these frames to the Physical Layer for actual transmission. Most link protocols add a certain amount of overhead at the beginning and end of the actual data to be sent. This overhead usually consists of an assigned number of the frame, frame type identifiers, frame source and/or destination identifiers, and some sort of mathematically derived number that is used as a check to make sure both sides of the physical interface have the same data. These basic functions are described in an ISO standard (ISO 3309), commonly referred to as the High-level Data Link Control protocol, or HDLC.

There are two versions of Link Layer protocols commonly used in amateur packet radio today. Both follow the HDLC standard for the addition of flags, address, control, and Frame Check Sequence (FCS) fields. The flags are used to indicate the beginning and end of the frame, the address field is used to indicate who the frame is from and/or going to, the control field is used to show what type of frame is being conveyed, and the FCS field is a cyclic-redundancy check calculated on the data between the opening and closing flags.

In order to assure t'he flag character (01111110) does not appear anywhere in a frame except at the beginning or end, anytime five or more one bits are found in the data, a zero bit is added after the fifth one bit. The receiving end will realize that the zero was added, and delete it.

The first thing most Link Layer protocols do is **to** establish a "virtual" connection between the two devices wishing to communicate. This allows both devices **to** know what mode each is in at any given time. In order to make and maintain this connection, certain types of frames are sent back and forth that don't carry any user data, but rather perform command or **sup**ervisory functions related to the status of the link. There are also supervisory link functions to make sure one device doesn't "overload" the other **with** data faster than the receiver can handle it.

Vancouver Protocol

The first Link Layer developed for use on the ham bands was based on the IBM variation of HDLC, called SDLC. This protocol was developed by Doug **Lockart, VE7APU,** the "father" of packet radio on the ham bands. It is connection oriented, and uses eight-bit address and control fields, along with the standard CRC for the FCS. There are a few supervisory frames necessary for creating and maintaining the connection, along with flow control frames to prevent overloading. The level 2 Vancouver protocol works fine, and its overhead is minimal.

AX.25 Level 2

After the AMRAD group used the Vancouver protocol for a while, it became obvious that there were some limitations to this protocol. The main limitation had to do with the addressing information imbedded in each frame. The Vancouver protocol uses eight bits for the addressing information. Some implementers of the Vancouver **protocol** modified it so that the addition of **"digital** repeaters" or digipeaters could be used. These additions took up two of the eight bits in the address field, leaving six bits for actual addressing. This meant that only 64 users could be addressed before overflow was reached. In addition, someone in each group had to assign these numbers to stations, and make sure that numbers weren't assigned twice.

AX.25 took care of this by **installing the** amateur's **callsign** in the address field. **One more** addition saw both the source and destination addresses in the address field. This meant that **the** address field of a frame jumped from one byte to 14 bytes in a **single** bound! A further addition saw **first** one, **and** now up to eight digital repeater addresses in the address field. Talk about overhead! Unfortunately, in order to design a system that hams can use easily, a system like this is almost a necessity.

In addition, AX.25 added more supervisory frames, and is designed to be more flexible in higher speed and full duplex systems. Most amateurs using packet radio today are using the AX.25 Level 2 standard, and all packet systems available today can support the AX.25 Level 2 protocol.

AX.25 also allows multiple link connections, so **that** several stations can be interconnected. This includes connecting to one's self, allowing testing of packet software if there are no other stations around (as long as there is a repeater available).

Those **wishing** to read more about these protocols should refer to the following:

Vancouver protocol available from:

Vancouver Amateur Digital Communications
Group **(VADCG)**
C/O Doug Lockhart**, VE7APU**
9531 Odlin **Road**
Richmond, B.C. **V6X 1E1**

AX.25 Level 2 protocol specification:

Second ARRL Amateur Radio Computer Networking Conference Proceedings available from the ARRL for $9.00.

Updates on the **AX.25** Level 2 protocol are available in the AMRAD Newsletter.

Digital Repeaters

Both the modified Vancouver protocol and the AX.25 Level 2 protocol support devices called **"digital** repeaters" or **"digipeaters".** These type of repeaters differ from **the** normal voice type repeater **in that they generally operate as time-**domain, or store-and-forward repeaters rather than the frequency-domain system used by voice systems. What this means is that a repeater will **listen to** a frequency for frames it should repeat. **When it hears** one, it **pulls it into its** memory, checking to be sure there are no **errors,** and then waits for the sender to drop its **transmitter.** The repeater then re-transmits the frame on the same frequency. This allows several packet stations to communicate over a single frequency that might not otherwise be able to hear each other. Since a single frequency is **used,** spectrum usage is **cut** in half. In addition, the repeater is usually a very simple device, since no cavities or filters are required.

Level 3. The Network Laver

The next level up the **ISO-RM** is the Network Layer. The units transferred at the **Network Layer are** called "packets". This level pr**obably should** have been split into two distinct leve**ls.** The lower level, sometimes called the Network Layer or Level **3A,** maintains control over a single, smaller network of users. The upper portion, called the **Internet** Layer **or** Level 3B, interconnects these **smaller** groups **into** a larger network, allowing **individuals** or **sy**stems in one group to communicate with others in **di**fferent groups **if** they want.

At this point, I think it would be advantageous to discuss for a moment the two basic types of network designs, the connection oriented, and the connectionless (clever name) or **Datagram type.** These two systems differ greatly in their design philosophy but either can be used in place of the other without adverse affects.

Some think that a whole network and internetwork must be the **same type**, or communications cannot **happen,** but with the **proper separation** of **functions,** gateways can **be built** allowing different systems at almost any level. A gateway is a **device that** transforms one type of protocol that exists on one side of it to **a** different **type protocol being** used at the other side of it. **When** properly **designed, gateways are** capable of interfacing two complete**ly diffe**rent style protocols to each other, as if the difference didn't exist.

Getting back to the two types of networks, **I will** first discuss the connection oriented network, followed by the connectionless type.

Connection Oriented Network

The connection oriented network operates very similarly to the Link Layer protocol. In order to transfer **any** user data across the network, a "connection" must first be made from one user to the other. This involves **passing** between the two stations (and any network controller that **may** exist) a connection request and acknowledgement. Once this connection is made, any data travelling between the two users must travel through the path established at the time the connection was created. If any unrecoverable errors occur, the defective connection must be torn down, and a new connection must be made, if possible.

Some of the advantages of a connection oriented protocol are:

1. Lower overhead per packet once a connection is made, since all information about who is communicating and what path is being used is sent only once (when the connection is being generated). This lower overhead usually simplifies the software necessary to implement the protocol.

2. **Out-of-sequence packets** generally aren't **allowed, again simplifying** the software needed **to** implement the network protocol, and also simplifying the higher leve**l protocols.**

3. Connection oriented protocols are generally easier to implement than datagram type protocols.

4. Once a connection is made, the routing of packets doesn't have to be recalculated over and over (and over and over) and over again.

Some of the disadvantages of the connection oriented network protocols are:

1. Since the route of data flow is established at connect time, if there is any failure along the path chosen, the connection must be torn down and re-established using a different path. This implies that any network using a connection oriented protocol should be as reliable as possible. Unreliable networks may take a longer amount of time to keep the network running than actually pass data.

2. If part of the network becomes overly congested, since there is no way to dynamically alter the path used in a connection, the congestion will become worse as time progresses, unless there is a way to automatically tear down and re-establish connections around the congested portion.

3. Out-of-sequence packets aren't normally allowed, causing Accurately received packets to be rejected because of badly received earlier packets. This could- cause an increase in channel occupation, reducing effective channel throughput.

4. If a station is moving through areas covered by connection oriented networks, it could have a problem when the time comes to leave one area and go into another. How a roving station can be passed from one network to another in connection oriented networks isn't a big problem presently, but it could become a problem as the use of packet radio increases.

There are more advantages and disadvantages for the connection oriented protocols, but those mentioned above are the most important.

Connectionless Protocols

The connectionless type of protocols (called the datagram type from here on) operate in a different manner than the connection type. In a datagram protocol, all information needed to get a packet from the source to its destination is included in the header of each packet. Obviously, this will cause the header to become larger than the equivalent packet of a connection oriented network. In addition, each packet's routing must be decided independantlv from others either preceeding or succeeding it, causing a lot of additional operating overhead while each packet switch decides the b&t way for this packet-to go. This can come in handy when a network is not too reliable, or when a portion of a network becomes congested, since the path taken by packets can be dynamically altered. This doesn't come cheaply however, it usually takes more computer power to make sure a datagram type network functions properly.

As the last paragraph illustrates, the advantages and disadvantages between connection oriented networks and datagram type networks are generally just the opposite of each other.

Level 3A. The Network Sublaver

The Network Sublayer is responsible for taking data from the higher level protocols, packetizing it, and sending it to the Link Layer for actual transmission through the Physical Layer. While the Link Layer is responsible for making sure the user data accurately transverses the physical link between two stations, the Network Layer is responsible for making sure that user data passes through any intervening nodes, such as metropolitan network controllers or packet switches. Along with this, the Network Layer makes sure that any data from another network either passes through the network successfully, or reaches the destination station if that station is part of the metropolitan network.

When I first began to study protocols above level 2, I was impressed by the datagram type of network. It seemed to have a lot going for it. especially in a relatively unstructured and potentially unreliable amateur radio packet network. Datagram networks are very forgiving by nature, allowing for the voluntary nature of amateur stations showing up, and then disappearing.

Then we found out how people were implementing datagrams, and on what type of machines. It seemed that most people were implementing datagrams on large computers or mini-computers. There didn't seem to be a practical implementation of a datagram network based on microcomputers.

In addition, the two major commercial data networks seemed to be implementing connection oriented networks very effectively, including the use of microcomputers in their implemenations. This is when I started taking a second look at the CCITT standard X.25, both at level 2 and level 3.

About the same time, Gordon Beattie Jr, N2DSY, was coming to the conclusion that X.25 would be a good place to start on establishing a standard protocol for levels 2 and 3. In the summer of 1982, Gordon came down to the Washington area, and we had a conference with Eric Scace, K3NA, at Telenet.

Eric became a valuable asset in our discussions, since in addition to working at Telenet, he served on the CCITT committee on X.25. It turns out that there can be a large difference between what a protocol document appears to say, and how the protocol is actually implemented. This is where Eric really helped, by giving us insight not only into w'hat the protocol designers meant, but also how the real world networks implemented the protocol.

As a result of these meetings, we came up with drafts of protocols for both levels 2 and 3. Level 2 eventually grew into the AX.25 Level 2 that most packeteers are now using. Level 3 is a much larger, more sophisticated protocol, and as such, requires more careful analysis to see what we need and what we don't in the amateur community. As with level 2, we can't just throw out portions of the protocol without making sure they wont be needed in the future.

A separate paper in these proceedings discusses the level 3 protocol in some detail, so I wont get describe it in detail here. It is based on the CCITT X.25 Level 3 protocol, using "virtual circuits". Permanent virtual circuits weren't deemed to be useful, at least at this point, in the amateur service, and the Datagram service of X.25 was eliminated by the CCITT because of lack of interest.

One of the main arguments used against connection oriented networks is that they aren't very forgiving in unreliable enviroments. It seems that most metropolitan networks should be reliable enough to support connections without major problems. Since connections require less channel overhead than datagrams, this should also allow more efficient use of RF frequencies.

The recommendation to go AX.25 at the Network Sublayer is not cast in stone, but it appears that this is the best compromise protocol to use at the local or metropolitan level.

Internet Sublayer

The Internet Sublayer is the next step (or half-step in this case> up the ladder to the user. This level isn't necessary for purely local or metropolitan communications, since the data at that level isn't intended to go outside the individual network. The Internet Sublayer is only necessary when data must flow outside a single network s boundry.

Since the Internet Sublayer is responsible for the transfer of data across individual networks to the destination network, there must be enough addressing information in the level 3B header to make sure the packet can be successfully routed to its destination. The internet protocol is also responsible for making

sure any fragmentation of large packets into smaller packets is done in an orderly fashion.

The amateur community is very inventive, and often likes to use whatever is invented locally rather than using a "standard" foisted on us by some outside group. Keeping this in mind, and also keeping in mind the potential of some networks being not as reliable as others, I propose that we use a datagram type of internet protocol. Even though datagrams might require more computer power to implement, not every user will be required to have this overhead, since the internet protocol is used to interconnect the individual networks, not each user.

Among the datagram type internet protocols available are the DARPA internet protocol and the National Bureau of Standards (NBS) internet protocol. These two are very similar, in fact the NBS standard grew out of the DARPA one. It seems that either of these might suit our needs with some slight "adjustments" for amateur peculiarities. The main difference between these two is that the NBS version has longer address fields (which we may need). The DARPA internet adds a minimum of 20 bytes of header (more for options), while the NBS version adds a minimum of 28 bytes. Otherwise, both look almost identical. Figure 2 shows the outline of an NBS internet header. Unfortunately, it is beyond the scope of this paper to discuss the operation of these protocols.

One important thing to keep in mind when discussing internet protocols, particularly the datagram type, is that the internet protocol must work very closely with the next level protocol, the Level 4, or Transport Layer protocol. The datagram type internet assumes that a rather large transport protocol resides above it, making sure that any alterations of data that might crop up due to internet operations (such as packets arriving out of sequence) are properly corrected. This interdependence is why the internet and transport levels are often referred to as one combination protocol (such as TCP/IP which means Transmission Control Protocol/Internet Protocol). It is important to keep this in mind when designing or implementing an internet protocol.

As mentioned before, just because a datagram type of protocol is chosen for the Internet Sublayer, this DOES NOT mean that a datagram Network Sublayer must also be implemented. This is just NOT true. In fact, included in the NBS documents on the NBS internet protocol is software describing an interface to an X.25 Network Sublayer. The two are separate items, and should be delt with as such.

Level 4, The Transport Layer

The main function of the Transport Layer is to make sure the data passed on from the higher levels at one side of a group of networks interconnected using an internet protocol is received at the intended destination correctly.

Part of this responsibility is to make sure the data is received in the same order as it was sent. In datagram protocols, it is possible for one packet sent before another to arrive at the destination network after the second one. This could cause big problems if left uncorrected. The Transport Layer must make sure all packets are in the correct order before sending them on up the ISO-RM ladder to the higher levels. This may involve buffering the packets for a period of time, potentially requiring large amounts of memory.

Another responsibility of the Transport Layer is to notify the originating network that the packet successfully reached the destination network. In addition, the Transport Layer may impose flow control procedures on packets as necessary.

As mentioned earlier, the Transport Layer works very closely with the Internet Sublayer. This means that if the DARPA internet is used, the DARPA transport protocol should also be used. The DARPA transport protocol adds an additional 20 bytes minimum of overhead as a transport header. If the NBS internet is chosen, the NBS transport protocol should also be implemented. The NBS version is more complicated than the DARPA version, and some of it might have to be thrown out if it is to be used on a microcomputer system.

Level 5, The Session Layer

Now that the data has transversed the network successfully, it is ready to be used by the intended destination device. If that destination device is a larger computer, capable of running several programs simultaneously, there must be a way of telling which program the received data is intended for. This is part of the responsibility of the Session Layer.

One example of this might be Dave, K8MMO's system having someone running an orbit prediction program the same time as another person is editing a document, both running under MP/M II. The other example might be having several -different people using the same program, such as a bulletin board program, at the same time.

The Session Layer adds its own overhead to make sure the proper application (be it a program or another user) gets the correct user data. The Session Layer introduces a new term for the block of data it deals with, the "message". Within the overhead that the Session Layer adds is some sort of routing information to insure that the data received from the network is sent to the proper program within the computer, which is referred to as a "port". These ports are generally assigned names by the application being run.

The Session Layer also makes sure that an operating session between a user at one end and the program at the other end is handled smoothly. If the user should suddenly disappear from the system, it is up to the Session Layer to inform the application of this problem, so that the application can take any action deemed necessary. This implies that the Session Layer not only handles data between the network (via the Transport Layer) and any applications involved, it also passes some status and control information between the network and the application in question.

The Session Layer is not a necessity in a lot of instances, such as two people typing back and forth ala RTTY mode. In this case, the Session Layer overhead could be considered unnecessary and eliminated.

The Session Layer is a subject that needs further study at this time, as there are several versions out (DARPA, NBS, CCITT. BX.25 etc). Since there aren't a lot of mainframes on the a packet networks so far (there isn't even a network as such), there is time to study this level carefully before making a commitment to any particular standard.

Level 6. The Presentation Laver

The Presentation Layer is responsible for making sure that the data passed from one end of a hook-up to the other end makes some sense, and is displayed in an orderly fashion. It specifies things such as what character code is used and screen and printer display control sequences (such as cursor addressing).

The Presentation Layer can be a very complicated system, or it can be a null level, depending on what type of devices are being used at each end.

If, for example, a glass TTY (such as is used for the hearing impaired) is to be used with a version of a word processor set for a Heath H-19 terminal, the Presentatin Layer would be very complicated. Not only would there be code conversions required (ASCII vs Baudot), but also screen formatting characters would have to be converted, along with other problems. The end that would do the conversion would depend on what type of Presentation Layer protocol had been agreed to by the users of the system.

If a different user was to use the same word processor with a Heath H-19, and the Presentation Layer protocol agreed to was the H-19 running ASCII, the Presentation Layer at both ends could end up being a null level, since the same protocol is implied at both ends.

Level 7, The Application Layer

Application Layer protocols are primarily concerned with how a particular program is operated by the user of the program. The application protocols are established so that users (be they individual or another program) will know how to correctly use a program through the network.

The Application Layer, being the top of the system, would normally be the last area to look at for standardization. Since there are a myriad of programs that could be run as application programs over the amateur packet radio network (and a lot more not even thought of, or written yet) this could end up being the hardest set of protocols to come up with.

Two types of programs that should have standard protocols written for fairly quickly though. They are the message system (generic name) and the file transfer programs.

There are many message system programs available to the amateur today. It seems that every one of these systems uses different commands to operate it, along with a different message format. It would help greatly if there could be a single, standard set of commands available, along with a standard message format. Then, each message system along a network could potentially access other message systems along the network, and automatically grab off any pertinent data. Also, there could then be defined within this message system protocol, a way of automatically forwarding messages along the network from a source message system to the destination message system.

There are many different message systems, and many different message system "standards" already in existance. DARPA has a standard, so does NBS, and the CCITT just ot name a few. This is an area I haven't delved into too far yet, so I have no feeling at this time as to which protocol would best suit our needs. Some initial work is being done by Paul Rinaldo, W4RI, Hank Magnuski, KA6M, Larry Kayser, WA3ZIA, along with the AMSAT and VITA contingent on message system standardization.

The other Presentation Layer protocol that needs almost immediate attention is the file transfer protocol. A lot of us are presently using the Ward Christensen protocol so prevalent among CP/M users for exchanging CP/M files using modems over the phone lines. In fact, this protocol has been implemented in many computer systems other than CP/M, including 6800 type computers and (rumor has it) DEC minis.

One of the faults with the CP/M file transfer protocol is that it uses a very simple checksum on the data transferred to make sure no errors crept into the transfer. There has been some modifications made in this area recently, some versions of this transfer program now allow either the original checksum routine or a more sophisticated CRC type calculation. Since there is so much redundant checking of data at the lower levels of a network, the more sophisticated version may not be needed.

There is also a protocol for file transfer floating around that was developed by the NBS but I haven't had a chance to study it carefully enough yet to see if it would fit our needs.

Conclusion

The OSI-RM appears to be taking shape in the amateur packet radio network. There is some protocol development work being done at almost all levels of the Reference Model, with most people working from the ground up at this point.

One of the disadvantages of the OSI-RM is thet there is a lot of added overhead, as mentioned at the beginning of this paper. This is primarily because multiplexing of different data paths is allowed at each level, causing multiple flow control procedures and addresses to be required at each level.

An alternative to the OSI-RM system might be to break the overall network design at different places. Eliminating the redundant capability of multiplexing of operations at each level would reduce the total amount of overhead required. This would have to be done very carefully.

It is hoped that this paper will help the newcomer to amateur packet radio understand how a data network is designed and implemented using the OSI Reference Model to allow a maximum of flexibility to the designers and implementers. I further hope that this paper stirs interest in the more advanced packet radio enthusiasts by stating my opinions and suggestions on recommendations at the various levels of the OSI-RM.

Comments or suggestions regarding any portion of this paper should be addressed to the author at the above address, or be sent to the Amateur Radio Research and Development (AMRAD) Newsletter for publication at the following address:

Amateur Radio Research and Development
PO Drawer 6148
McLean VA 22106-6148

```
                                   Octets
                         1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3
     1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    ---------------------------------------------------------------
    ! Version !  IHL  !  Grade of Service  !     Data Unit 'Length    !
    ---------------------------------------------------------------
    !            Identifier                !Flags !   Fragment offset !
    ---------------------------------------------------------------
    !     Lifetime    !  User Protocol     !      Header Checksum     !
    ---------------------------------------------------------------
    !                         Source Address                          !
    ---------------------------------------------------------------
    !                    Source Address (continued)                   !
    ---------------------------------------------------------------
    !                       Destination Address                       !
    ---------------------------------------------------------------
    !                 Destination Address (continued)                 !
    ---------------------------------------------------------------
    !              Options as needed             !      Padding       !
    ---------------------------------------------------------------
    !                              Data                               !
    !               Note: This field is of variable length            !
    !                           End of Data                           !
    ---------------------------------------------------------------
```

Figure 2. NBS Internet Header Format

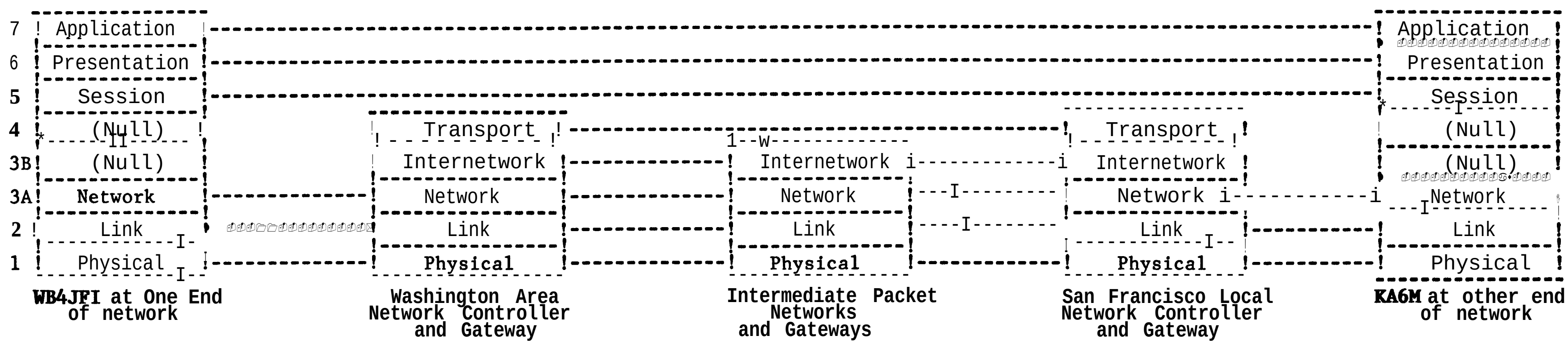


Figure 1. Example of ISO Interconnections in an Amateur Network
Shown is a Theoretical Packet Connection Between WB4JFI in the Washington DC Area, and KA6M in the San Francisco Area

AX.25 NETWORK **SUBLAYER** PROTOCOL RECOMMENDATION

Terry Fox, WB4JFI
President, AMRAD
1819 Anderson Rd.
Falls Church, VA 22043

Introduction

This is the first of four papers that make up a protocol recommendation for AX.25 at Level 3A, the Network Sublayer.

This series of papers is being generated by AMRAD after a series of meetings between AMRAD members Paul Rinaldo, W4RI, Terry Fox, WB4JFI, Dave Borden, K8MMO, Eric Scace, K3NA, and Gordon Beattie, N2DSY.

These papers are first drafts, and are being released to the amateur community for comments and suggestions. Anyone wishing to comment is invited to write to the author at the above address, or write to the AMRAD Newsletter at the following address:

Amateur Radio Research and Development Corp.
PO Drawer 6148
McLean, VA 22186-6148

The protocol recommendation that follows is based on the CCITT X.25 Level 3 specification. Since many amateurs may not have the CCITT documents available to them, it was decided to replicate the entire document, with additions or deletions necessary to apply the protocol to the amateur enviroment.

This paper will discuss some of the basics of the Network Sublayer, along with operating procedures. The next paper will describe the actual packet formats. The third in the series will describe optional user requested facilities, and the fourth paper will include the Annexes mentioned throughout the previous three papers.

2 AX.25 Network Sublayer Recommendation

3.0.1 Network Sublayer Basics

The Network Layer of the IS0 Reference Model is generally broken into two different parts, each responsible for separate, distinct functions.

The Network Sublayer is responsible for the proper operation of a local, or metropolitan group of interconnected packet devices. These interconnected devices make up a network of packet users, who wish to communicate either with each other, or possibly with others outside of the metropolitan network. How the data is transferred outside of the metropolitan network is a function of the Internetwork Sublayer, and as such, falls outside the domain of this recommendation.

The Network Sublayer relies on a lower level protocol (usually the AX.25 Link Layer protocol) to cause data at the Network Layer to be transferred from one device to another. While the Link Layer protocol is responsible for the user data to transverse a physical medium properly, the Network Sublayer is responsible for the accurate transfer of data through the metropolitan network. This is usually accomplished by having individual devices interconnected to a master device (called a network controller, network hub, packet switch, or DCE). Any packet user wishing to communicate with another user (either within the metropolitan network, or outside it) does this through this master device.

Subject for further study is how two stations communicate at the Network Sublayer in the absence of a packet switch. One proposal is to have the two stations arbitrate which one will become the packet switch by a simple comparison of callsigns.

3.0.4 Network Sublayer Responsibilities

The Network Sublayer is responsible for taking data from the 'higher level protocols and sending that data to the intended destination device, through the lower level protocols that may be implemented.

Since the recommended protocol is connection oriented, in order to pass data over the network, a "virtual" connection must first be made with the destination device through a network controller, or packet switch. This recommendation handles the establishment, proper operation, recovery from errors, and tearing down of these connections necessary to pass data, along with the actual data transmissions.

This protocol is also responsible for passing along certain status information of the network or lower levels to the higher level protocols.

3.0.5 Device Descriptions

At the Network Sublayer there are two types of devices presently defined. Their descriptions have been adjusted slightly to take into account the amateur enviroment. Neither one of these devices are usually "real" devices, rather they are usually a software implemented "machine".

3.0.5.1 Data Terminating Equipment

At the Network Sublayer, the Data Terminating Equipment device, hereafter called the DTE, is generally considered the individual packet user, be it an actual user, a remote network gateway, or a large computer device running programs available to other users.

3.0.5.2 Data Circuit-Terminating Equipment

At the Network Sublayer, the Data Circuit-Terminating Equipment 'Equipment device, hereafter called the DCE, is either the packet switch device or if there is no packet switch device available, one of the DTE devices wishing to communicate. The arbitration of the second possibility is undefined at this point, and is a subject for further study.

3.0.6 Units of Data Transferred at the Network Sublayer

The basic unit of data that is transferred across a DTE/DCE interface is called a "packet". This packet is contained within the information field of Link Layer frames. All packets must contain an integral number of octets. In addition, the data field of data packets must also contain an integral number of packets.

3.0.7 Types of connections available

Amateur X.25 defines only one type of connection at the metropolitan network level, that of the virtual call. Permanent virtual circuits and datagrams are not supported by by AX.25 as presently defined.

3.1 Logical Channels

To enable multiple simultaneous virtual calls to exist, logical channels are used. Each virtual call is assigned a logical channel group number (LCGN) (less than or equal to 15 decimal)

and a logical channel number (LCN) (less than or equal to 255 decimal). A logical channel group number and a logical channel number is assigned during the call set-up phase. The actual range of logical channel numbers available is established by the network, and agreed upon at the time of connection to the network. Annex A shows the recommendation of LCGN and LCN values used.

3.2 Basic Structure of Packets

Every packet transferred across the DTE/DCE interface consists of at least three octets. Within these three octets are a general format identifier, a logical channel identifier, and a packet type identifier. Additional packet fields may be appended as appropriate. Packet types are shown in Table 5/AX.25.

From DCE to DTE	From DTE to DCE
Call set-up and clearing	
Incoming call	Call request
Call connected	Call accepted
Clear indication	Clear request
DCE clear confirmation	DTE clear confirmation
Data and interrupt	
DCE data	DTE data
DCE interrupt	DTE interrupt
DCE interrupt confirmation	DTE interrupt confirmation
Flow control and reset	
DCE RR	DTE RR
DCE RNR	DTE RNR
Reset indication	Reset request
DCE reset confirmation	DTE reset confirmation
Restart	
Restart indication	Restart request
DCE restart confirmation	DTE restart confirmation
Diagnostic	
Diagnostic	

Table 5/AX.25 AX.25 Packet types

3.3 Procedure for Restart

The restart procedure is used to initialize or re-initialize the network level DTE/DCE interface. The restart procedure clears all virtual calls at the DTE/DCE interface.

3.3.1 Restart by the DTE

The DTE may at any time request a restart by sending a restart request packet across the packet interface. This will then place each logical channel in the DTE restart request state (r2).

The other device (DCE or possibly another DTE) will confirm the restart by sending a DCE restart confirmation packet and placing the logical channels used between the two devices in the ready state (p1).

The DCE restart confirmation packet only applies to virtual calls between the requesting DTE and the receiving DCE. This restart has no affect on virtual calls with any other device. Time spent in the DTE restart request state (r2) shall not exceed time-limit T20 (see Annex D).

3.3.2 Restart by the DCE

The DCE may initiate a restart of the packet interface by sending a restart indication packet. All virtual calls for the specified packet interface are then placed in the DCE restart indication state (r3). While in this state the DCE will ignore all packets received from the DTE involved except for restart request and DTE restart confirmation packets.

The DTE will confirm the restart by sending a DTE restart confirmation packet, and then place all virtual calls between it and the DCE in question in the ready state (p1).

The action taken by the DCE when the DTE does not confirm restart within time-out T10 is given in Annex D.

3.3.3 Restart Collision

Restart collision occurs when both the DCE and DTE simultaneously send a restart request and a restart indication packet. When this happens, the DCE will consider the restart completed. The DCE will not expect a DTE restart confirmation packet and will not send a DCE restart confirmation packet. This places all virtual calls between the affected DTE/DCE interface in the ready state (p1).

3.4 Error Handling

Table C-1/AX.25 indicates the reaction of the DCE when certain error conditions are encountered. Error conditions other than those specified in the table are discussed in sections 4 and 5.

3.4.1 Diagnostic Packet

The diagnostic packet may be used to indicate error conditions when the usual methods (ex. reset, clear and restart with cause and diagnostic) are inappropriate. A diagnostic packet from a DCE should provide information on the error situations that are considered unrecoverable at the packet layer. This information permits analysis of the error, and recovery by higher levels at the DTE, if possible.

A diagnostic packet is issued only once per particular instance of an error condition. A DTE receiving a diagnostic packet is not required to confirm reception. After a DCE sends a diagnostic packet, it will same in the same state for that logical channel(s) as it was when the diagnostic was generated.

3.5 Effects of the Physical and Link Layer on the Packet Level

Changes of operational states of the Physical and Link Layers do not implicity alter the state of each logical channel at the packet level. When changes that affect the packet level do occur, they are explicitly indicated at the packet level by the use of restart., clear, or reset procedures, whichever is appropriate.

A failure at the Physical and/or Link Layers is defined to be when the DCE cannot transmit or receive any frames because of abnormal conditions at the Physical and/or Link Layer.

When a failure on the Physical and/or Link Layer is detected, virtual calls will be cleared, and further action may be taken as described in section 4.6.

When the failure of the Physical and/or Link Layer is recovered, the DCE will send a restart indication packet with the cause "Network operational" to the DTE. Any further action to be taken is defined in section 4.6.

In other out-of-order conditions on the Physical and/or Link Layer, the DCE will clear all virtual calls using the affected link.

Note: An out-of-order condition on the Link Level includes reception of a disc command or a transmission of a disc command by the DCE, in the case of a single link procedure.

4 Procedures for virtual circuit services

4.1 Procedures for virtual call service

Figures B-1/AX.25, B-2/AX.25, and B-3/AX.25 in Annex B show the state diagrams which give a definition of events at the packet level DTE/DCE interface for each logical channel used for virtual calls.

Annex C gives details of the action taken by the DCE on reception of packets in each state shown in Annex B.

The call set-up and clearing procedures described in the following paragraphs apply independently to each logical channel assigned to the virtual call service at the DTE/DCE interface.

4.1.1 Ready state

If there is no call in existance, a logical channel is in the ready state (p1).

4.1.2 Call request packet

The calling DTE shall indicate a call request by sending a call request packet across the DTE/DCE interface. The logical channel selected by the DTE is then in the DTE waiting state (p2). The call request packet includes the called DTE address. The calling DTE address shall also be used.

Note 1. A DTE address shall be encoded as described in Annex F of this document.

Note 2. In order to minimize the risk of call collisions, the call request packet should use the logical channel with the highest number in the range allowed in Annex A that is in the ready state.

4.1.3 Incoming call packet

The DCE will indicate that there is an incoming call by sending across the DTE/DCE interface an incoming call packet. This will place the logical channel in the DCE waiting state (p3).

The incoming call packet will use the logical channel in the ready state with the lowest number in the range allowed in Annex A. The incoming call packet shall include the DTE calling address and the called DTE address fields encoded as described in Annex F.

4.1.4 Call accepted packets

The called DTE shall indicate its acceptance of the call by sending a call accepted packet across the DTE/DCE interface. This call accepted packet will specify the same logical channel as that of the incoming call packet. This places the specified logical channel in the data transfer state (p4).

If the called DTE does not accept the call by sending a call accepted packet or does not reject it by sending a clear request packet as described in paragraph 4.1.7 within time-out T11 (see Annex D), the DCE will consider it as a procedure error from the called DTE and will clear the virtual call aceording to the procedure described in paragraph 4.1.8.

4.1.5 Call connected packet

The reception of a call connected packet by the calling DTE specifying the same logical channel as that specified in the call request packet indicates that the call has been accepted by the called DTE by means of a call accepted packet. This places the specified logical channel in the data transfer state (p4).

The time spent in the DTE waiting state (p2) will not exceed time-out T21 (see Annex D).

4.1.6 Call collision

Call collision occurs when a DTE and DCE simultaneously send a call request packet and an incoming call packet specifying the same logical channel. The DCE will proceed with the call request and cancel the incoming call.

4.1.7 Clearing by the DTE

The DTE may indicate clearing at any time by sending a clear request packet across the DTE/DCE interface (see paragraph 4.5). The logical channel is then in the DTE clear request state (p6). When the DCE is prepared to free the logical channel, it will send a clear confirmation packet across the DTE/DCE interface specifying the proper logical channel. This logical channel is then placed in the ready state (p1).

The DCE clear confirmation packet has only local significance, it does not affect calls outside the one logical channel cleared (such as end-to-end calls). The time spent in the DTE clear request state (p6) will not exceed time limit T23 (see Annex D).

It is possible that subsequent to sending a clear request packet and prior to the reception of a DCE clear confirmation packet, the DTE will receive other types of packets (depending of the state of the logical channel).

The calling DTE may abort a call by clearing it before it has received a call connected or clear indication packet.

The called DTE may refuse an incoming call by clearing it as described above instead of sending a call accepted packet.

4.1.8 Clearing by the DCE

The DCE will indicate clearing by transmitting across the DTE/DCE interface a clear indication packet (see 4.5). The logical channel is then in the DCE clear indication state (p7). The DTE shall respond by sending a DTE clear confirmation packet. The logical channel is then placed in the ready state (p1).

The action taken by the DCE when the DTE does not confirm clearing within time-out T13 is given in Annex D.

4.1.9 Clear collision

Clear collision occurs when a DTE and DCE simultaneously send a clear request packet and a clear indication packet specifying the same logical channel number. When this happens, the DCE will consider the clearing completed and will not expect a DTE clear confirmation packet. The DCE will not send a DCE clear confirmation packet. The logical channel will be placed in the ready state (p1).

4.1.10 Unsuccessful call

If a call cannot be established, the DCE will send a clear indication packet specifying the logical channel indicated in the call request packet to the calling DTE.

4.1.11 Call progress signals

The DCE will be capable of transferring to the DTE clearing progress signals as specified in a future document (AX.96).

Clearing call progress signals will be carried in clear indication packets which will terminate the call to which the packet refers. The method of coding clear indication packets containing call progress signals is detailed in paragraph 6.2.3.

4.1.12 Data transfer state

The procedures for the control of packets between DTE and DCE while in the data transfer state are contained in section 4.3 below.

4.3 Procedures for data and interrupt transfer

The data transfer and interrupt procedures described in the following paragraphs apply independantly to each logical channel assigned for virtual calls existing at the DTE/DCE interface.

Normal network operation dictates that user data in data and interrupt packets are all passed transparently, unaltered through the network in the case of packet DTE to packet DTE communications. The order of bits in data packets is preserved. Packet sequences are delivered as complete packet sequences. Diagnostic codes are treated as described in sections 6.2.3, 6.5.3, and 6.6.1.

4.3.1 States for data transfer

A virtual call logical channel is in the data transfer state (p4) after completion of call establishment and prior to a clearing or restart procedure. Data, interrupt!, flow control, and reset packets may be transmitted and received by a DTE in the data transfer state of a logical channel at the DTE/DCE interface. In this state, the flow control and reset procedures described in

section 4.4 apply to data transmission on that logical channel to and from the DTE.

When a virtual call is cleared, data and interrupt packets may be discarded by the network (see 4.5). In addition, data, interrupt, flow control, and reset packets transmitted by a DTE will be ignored by the DCE when the logical channel is in the DCE clear indication state (p7). It is left to the DTE to define DTE to DTE protocols able to cope with various possible situations that may occur.

4.3.2 User data field length of data packets

The standard maximum user data field length is 128 octets.

The user data field of data packets transmitted by a DTE or DCE may contain any number of octets up to the agreed upon maximum. The user data field shall contain an integral number of octets.

If the user data field in a data packet exceeds the maximum user data field length, the DCE will reset the virtual call with the resetting cause "Local procedure error".

4.3.3 Deliverv confirmation bit

The setting of the Delivery Confirmation bit (D bit) is used to indicate whether or not the DTE wishes to receive an end-to-end acknowledgement of delivery of the packet with the D bit set. This acknowledgement is for data it has transmitted, and the acknowledgement is made using the packet receive sequence number P(R) (see 4.4 below).

The use of the D bit does not obviate the need for a higher level protocol between communicating DTEs which may be used independantly of the D bit procedure to recover from user or network generated resets and clearings.

4.3.4 More data mark

If a DTE or DCE wishes to indicate a sequence of more than one packet, it uses the More Data Mark bit (M bit) as defined below.

The M bit can be set to one in any data packet. When it is set to one in a full data packet, or in a partially full data packet also carrying the D bit set to one, it indiactes that more data is to follow. Networks supporting AX.25 will not perform data packet segmentation or recombination.

A sequence of data packets with every M bit set to one except the last one will be delivered as a sequence of data packets with the M bit set to one except for the last one when the original packets having the M bit set to one are either full (irrespective of the setting of the D bit) or partially full but have the D bit set to one.

Two catgories of packets, A and B have been defined as shown in Table 6/AX.25. Table 6/AX.25 also illustrates the networks treatment of the M and D bits at both ends of a virtual call.

Data packet sent by source DTE				Data packet rcvd by the destination DTE	
Category	M	D	Full	M	D
B	0 or 1	0	No	0	0
B	0	1	No	0	1
B	1	1	No	1	1
B	0	0	Yes	0	0
B	0	1	Yes	0	1
A	1	0	Yes	1	0
B	1	1	Yes	1	1

Table 6/AX.25
Definition of two categories of data packets and network treatment of the M and D bits

4.3.5 Complete packet sequence

A complete packet sequence is defined as being composed of a single category B packet and all contiguous preceeding category A packets having the exact maximum user data field length with the M bit set to one and the D bit set to zero. All other data packets are category B packets.

When transmitted by a DTE source, a complete packet sequence is always delivered to the destination DTE as a single complete packet sequence.

The user data field of the last packet of the sequence may have less than the maximum length and the M and D bits are set as described in Table 6/AX.25.

Since the maximum user data field length is the same at both ends, the user data fields of data packets are delivered to the receiving DTE exactly as they have been received by the network. If the last packet of a complete packet sequence transmitted by the source DTE has a data field less than the maximum length with the M bit set to one and the D bit set to zero, then the last packet of the complete packet sequence delivered to the receiving DTE will have the M bit set to zero.

4.3.6 Qualifier bit

A complete packet sequence may be on one of two levels. If a DTE wishes to transmit on more than level, it uses the Qualifier bit (Q bit).

When only one level of data is being sent on a logical channel, the Q bit is always set to zero. If two levels of data are being sent, the transmitting DTE should set the Q bit in all data packets of a complete packet sequence to the same value, either zero or one. A complete packet sequence, which is sent with the Q bit set to the same value in all packets, is delivered by the network as a complete packet sequence with the Q bit set in all packets to the value assigned by the transmitting DTE.

When the Q bit is not set to the same value by the transmitting DTE within a complete packet sequence? a network supporting AX.25 will reset the logical channel with the cause "Local procedure error" and a diagnostic code of "Inconsistant Q bit setting".

Recommendation AX.29 gives an example of the procedures to be used when the Q bit is set to one.

Packets are numbered consecutively (see 4.4.1.1) regardless of their data level (Q bit setting).

4.3.7 Interrupt procedure

The interrupt procedure is used to allow a DTE to transmit data to the remote DTE without following the flow control procedure applying to data packets (see 4.4). The interrupt procedure applies only in the flow control ready state (d1) within the data transfer state (p4).

The interrupt procedure will have no effect on the transfer and flow control procedures applying to the data packets on a virtual call logical channel.

To transmit an interrupt, the DTE sends a DTE interrupt packet across the DTE/DCE interface. The DTE should not send a second DTE interrupt packet until the first one is confirmed by the reception of a DCE interrupt confirmation packet (see Note 2 to Table C-4/AX.25). After the interrupt procedure is completed at the remote end, the DCE will confirm the receipt of the interrupt by sending a DCE interrupt confirmation packet. The reception of a DCE interrupt confirmation packet indicates that the interrupt has been confirmed by the remote DTE by means of a DTE interrupt confirmation packet.

The DCE indicates an interrupt from the remote DTE by sending a DCE interrupt packet across the DTE/DCE interface which contains the same data field as that in the DTE interrupt packet transmitted by the remote DTE. A DCE interrupt packet is delivered at or before the point in the data packets stream at which the DTE interrupt packet was generated. The DTE will

confirm reception of the DCE interrupt packet by sending a DTE interrupt confirmation packet.

4.4 Procedures for flow control

Section 4.4 only applies to the data transfer state (p4) and specifies the procedures covering flow control of data packets and reset on each logical channel used for a virtual call.

4.4.1 Flow Control

The transmission of data packets across a DTE/DCE interface of a logical channel in a virtual call is controlled separately for each direction, and is based on authorization from the receiver.

Flow control also allows a DTE to limit the rate at which it accepts packets across the DTE/DCE interface, noting that there is also a network-dependant limit on the number of packets which may be within the network for the virtual call.

4.4.1.1 Numbering of data packets

Each data packet transferred across the DTE/DCE interface for each direction of transmission in a virtual call is numbered sequentially.

The sequence numbering scheme of the packets is in module 8. The packet sequence numbers cycle through the entire range from zero to seven. The packet sequence numbering scheme is the same for both directions of transmission and is common for all logical channels at the DTE/DCE interface.

Only data packets contain this sequence number, which is called the packet send sequence number P(S).

The first data packet sent across the DTE/DCE interface for each direction of data transmission when the logical channel has just entered the flow control ready state (dl), will have a packet send sequence number equal to zero.

4.4.1.2 Window description

For each direction of a data transmission over a virtual call logical channel, a window is defined as the ordered set of W consecutive packet send sequence numbers of the data packets authorized to cross the interface.

The lowest sequence number in the window is referred to as the lower edge. When a virtual call at the DTE/DCE interface has just entered the flow control ready state (dl), the window related to each direction of data transmission has a lower window edge equal to zero.

The packet send sequence number of the first data packet not authorized to cross the interface is the value of the lower window edge plus W (module 8).

The standard window size W is 2 for each direction of data transmission at the DTE/DCE interface.

4.4.1.3 Flow control principles

When the sequence number P(S) of the next packet to be sent by the DCE is within the window, the DCE is authorized to transmit this data packet to the DTE. When the sequence number P(S) of the next data packet to be transmitted by the DCE is outside of the window, the DCE shall not transmit a data packet to the DTE. The DTE should follow this same procedure.

When the sequence number P(S) of the data packet received by the DCE is the next in sequence and is within the window, the DCE will accept the data packet. A received data packet containing a P(S) that is out of sequence (such as when there is a gap in the received P(S) numbering, or a duplicate P(S) number), out of window, or not equal to zero for the first data packet after entering the flow control ready state (dl) is considered by the DCE as a local procedure error. The DCE will reset the virtual call (see 4.4.3). The DTE should follow the same procedure.

A number (modulo S), referred to as a packet receive sequence number P(R), conveys across the DTE/DCE interface information from the receiver for the transmission of data packets. When transmitted across the DTE/DCE interface, a P(R) becomes the lower window edge. In this way, additional data packets may be authorized by the receiver to cross the DTE/DCE interface.

The packet receive sequence number, P(R), is conveyed in data, receive ready (RR), and receive not ready (RNR) packets.

The value of a P(R) received by the DCE must be within the range from the last P(R) received by the DCE up to and including the packet send sequence number of the next data packet to be transmitted by the DCE. Otherwise, the DCE will consider the reception of this P(R) as a procedure error and will reset the virtual call. The DTE should follow the same procedure.

The receive sequence number P(R) is less than or equal to the next data packet sequence number expected, and implies that the DTE or DCE transmitting P(R) has accepted at least all data packets numbered up to and including P(R)-1.

4.4.1.4 Delivery confirmation

When the D bit is set to zero in a data packet having P(S) equal to p, the significance of the returned P(R) corresponding to the data packet (ex. P(R)=p+1) is a local updating of the window across the packet level interface so that the achievable throughput is not constrained by the DTE-to-DTE round trip delay across the network(s).

When the D bit is set to one in a data packet having P(S)=p, the significance of the returned P(R) corresponding to that data packet (ex. P(R)=p+1) is an indication that a P(R) has been received from the remote DTE for all data bits in the data packet in which the D bit had originally been set to one.

When a DTE receives a data packet with the D bit set to one, it should transmit the corresponding P(R) as soon as possible in order to avoid the possibility of deadlocks (without waiting for further data packets). A data, RR, or RNR packet may be used to convey the P(R) (see Note to 4.4.1.6). Likewise, the DCE is required to send P(R) to the DTE as soon as possible after it is received from the remote DTE.

In the case where a P(R) for a data packet with the D bit set to one is outstanding, the local updating of the window will be deferred for subsequent data packets with the D bit set to zero.

4.4.1.5 DTE and DCE receive ready (RR) packets

RR packets are used by the DTE or DCE to indicate that it is ready to receive W data packets within the window starting with P(R), where P(R) is indicated in the RR packet.

4.4.1.6 DTE and DCE receive not ready (RNR) pack-

RNR packets are used by the DTE or DCE to indicate a temporary inability to accept additional data packets for a given virtual call. A DTE or DCE receiving an RNR packet shall stop transmitting data packets on the indicated logical channel, but the window is updated by the P(R) value of the RNR packet. The receive not ready condition is cleared by the transmission in the same direction of a RR packet or by a reset procedure being initiated.

The transmission of a RR packet after a RNR packet at the packet level is not to be taken as a demand for retransmission of packets which have already been transmitted.

The RNR packet may be used to convey across the DTE/DCE interface the P(R) value corresponding to a data packet which had the D bit set to one in the case that additional data packets cannot be accepted.

4.4.2 Throughput characteristics and throughput classes

The attainable throughput on virtual calls carried at the DTE/DCE interface may vary due to the stastical sharing of transmission and switch resources and is constrained by:

1) the access line characteristics, local window size and traffic characteristics of other logical channels at the local DTE/DCE interface;

2) the access line characteristics, local window size and traffic characteristics of other logical channels at the remote DTE/DCE interface, and;

3) the throughput achievable on the virtual call through the network(s) independant of interface characteristics including number of active logical channels. This throughput may be dependant on network service characteristics such as window rotation mechanisms and/or optional user facilities requested on national/international calls.

The attainable throughput will also be affected by:

1) the receiving DTE flow controlling the DCE;

2) the transmitting DTE not sending data packets which have the maximum data field length;

3) the local DTE/DCE window and/or packet sizes, and ;

4) the use of the D bit.

A throughput class for one direction of transmission is an inherent characteristic of the virtual call related to the amount of resources allocated to this virtual call. This characteristic is meaningful when the D bit is set to zero in data packets. It is a measure of the throughput that is not normally exceeded on the virtual call. However, due to the statistical sharing of transmission and switching resources, it is not guaranteed that the throughput class can be reached 100% of the time.

Depending on the network and the applicable conditions at the considered moment, the effective throughput may exceed the throughput class.

The definition of throughput class as a grade of service parameter is for further study. The grade of service might be specified when the D bit is set to zero or over a time period between the completion and initiation of successive D bit procedures.

The throughput class can only be reached if the following conditions are met:

a) the access data links of both ends of a virtual call are engineered for the throughput class;

b) the receiving DTE is not flow controlling the DCE such that the throughput class is not attainable;

c) the transmitting DTE is sending data packets which have the maximum data field length, and;

d) all data packets transmitted on the virtual call have the D bit set to zero.

The throughput class is expressed in bits per second. At the DTE/DCE interface, the maximum data field length is specified for a virtual call, and thus the throughput class can be interpreted by the DTE as the number of full data packets per second that the DTE does not have a need to exceed.

The default throughput classes for both directions of transmission correspond to the user class of service of the DTE (see 7.4.2.6) but do not exceed the maximum throughput class supported by the network.

The summation of throughput classes of all virtual calls supported at a DTE/DCE interface may be greater than the data transmission rate of the access line.

4.4.3 Procedure for reset

The reset procedure is used to re-initialize the virtual call, and in so doing removes in each direction all data and interrupt packets which may be in the network (see 4.5). When a virtual call at the DTE/DCE interface has just been reset, the window related to each direction of data transmission has a lower window edge equal to zero, and the numbering of subsequent data packets to cross the DTE/DCE interface for each direction of data transmission shall start from zero.

The reset procedure can only apply in the data transfer state (p4) of the DTE/DCE interface. In any other state, the reset procedure is abandoned. As an example, when a clearing or restarting procedure is initiated, reset requested and reset indication packets can be left unconfirmed.

For flow control, there are three states (d1, d2, and d3) within the data transfer state (p4). They are flow control ready (d1), DTE reset request (d2), and DCE reset indication (d3) as shown in the state diagram in Figure B-3/AX.25. When entering state p4, the logical channel is placed in state d1. Table B-4/AX.25 specifies actions taken by the DCE on the reception of packets from the DTE.

4.4.3.1 Reset request packet

The DTE shall indicate a request for reset by transmitting a reset request packet specifying the logical channel. This places the logical channel in the DTE reset request state (d2).

4.4.3.2 Reset indication packet

The DCE shall indicate a reset sending to the DTE a reset indication packet specifying the logical channel and the reason for the resetting. This places the logical channel in the DCE reset indication state (d3). In this state, the DCE will ignore all data, interrupt, RR, and RNR packets.

4.4.3.3 Reset Collision

Reset collision occurs when a DTE and a DCE simultaneously transmit a reset request packet and a reset indication packet specifying the same logical channel. Under these circumstances the DCE will consider the reset completed. The DCE will not expect a DTE reset confirmation packet and will not transfer a DCE reset confirmation packet. This places the logical channel in the flow control ready state (d1).

4.4.3.4 Reset confirmation packets

When the logical channel is in the DTE reset request state (d2), the DCE will confirm reset by sending to the DTE a DCE reset confirmation packet. This places the logical channel in the flow control ready state (d1).

The reset confirmation packet has only local significance. The time spent in the DTE reset request state (d2) will not exceed time limit T22 (see Annex D).

When the logical cahnnel is in the DCE reset indication state (d3), the DTE will confirm reset by transmitting to the DCE a DTE reset confirmation packet. This places the logical channel in the flow control ready state (d1). The action taken by the DCE when the DTE does not confirm the reset within time-out T12 is given in Annex D.

4.5 Effects of clear, reset and restart procedures on the transfer of packets

All data and interrupt packets generated by a DTE (or the network) before initiation by the DTE or the DCE of a clear, reset, or restart procedure at the logical interface will either be delivered to the remote DTE before the DCE transmits the corresponding indication on the remote interface, or be discarded by the network.

No data or interrupt packets generated by a DTE (or the network) after the completion of a reset procedure at the local interface will be delivered to the remote DTE before the completion of the corresponding reset procedure at the remote interface.

When a DTE initiates a clear, reset, or restart procedure on its local interface, all data and interrupt packets which were generated by the remote DTE (or the network) before the corresponding indication is transmitted to the remote DTE will be either delivered to the initiating DTE before DCE confirmation of the initial clear, reset, or restart request, or be discarded by the network.

The maximum number of packets which may be discarded is a function of network end-to-end delay and throughput characteristics and, in general, has no relation to the local window size. For virtual calls on which all data packets are transferred with the D bit set to one, the maximum number of packets which may be discarded in one direction of transmission is not larger than the window size of the direction of transmission.

4.6 Effect of physical and link level failures

When a failure on the physical and/or link level is detected, the DCE will transmit to the remote end a clear with the cause "Out of order" for each existing virtual call.

During the failure, the DCE will clear any incoming virtual calls with the cause "Out of order" and a diagnostic code of "Call setup or clearing problem".

When the failure is recovered on the physical and link levels, the restart procedure will be auctioned (see 3.5).

5 Datagram service

At this time, datagram service is not available in AX.25.

PACKET FORMATS OF AX.25 LEVEL 3 PROTOCOL

Terry Fox, WB4JFI
President, AMRAD
1819 Anderson Rd.
Falls Church, VA 22043

Description

This paper is part two of a series of papers that describe The Network Sublayer portion of an AX.25 data communications system.

The purpose of this paper is to describe the formats of the various types of packets used to establish, maintain, and tear down a connection between a DTE and a DCE, along with the packets necessary to control the data flow along that connection while it is operational.

This paper was generated by taking the CCITT X.25 document and adding or deleting information pertaining to amateur radio data networking.

This is a first draft, corrections and ammendments will be forthcoming. Follow the AMRAD Newsletter (see information in the first paper of this series) for further information.

6 Packet Formats

6.1 General

The possible extension of packet formats by the addtion of new fields is a subject for further study. In general any additional field would:

a) only be provided as an addition following all previously defined fields, not as an insertion between any prevrously defined fields;

b) be transmitted to a DTE only when either the DCE has been informed that the DTE is able to interpret this field and act accordingly, or when the DTE can ignore the field without adversely affecting the operation of the DTE/DCE interface;

c) not contain any information pertaining to a user facility to which the DTE has not subscribed, unless the DTE can ignore the facility without adversely affecting the operation of the DTE/DCE interface.

Bits of an octet are numbered 8 to 1 where bit 1 is the low order bit and is transmitted first Octets of a packet are consecutively numbered starting at 1 and are transmitted in this order.

6.1.1 General format identifier

The general format identifier (GFI) field is a four bit binary coded field which is used to indicate the general format of the rest of the packet header. The GFI is located in bit positions 8, 7, 6, and 5 of octet 1, with bit 5 being the low order bit (see Table 7/AX.25). Values for the GFI not specified in Table 7/AX.25 are reserved for future use.

Bit 8 of the GFI is used for the Qualifier bit (Q bit) in data packets, and is set to 0 in all other packets.

Bit 7 of the GFI is used for delivery confirmation (D bit) in data packets. It is set to 1 in call set-up packets, and is set to 0 in all other non-data packets.

Bits 6 and 5 are encoded 0 and 1 respectively, indicating that all sequence numbering will be done modulo 8. The encoding of bits 6 and 5 to 1 and 0 is reserved for future use. Bit 6 and 5 encodings of both zero or both one are not allowed under this recommendation.

General Format Identifier (GFI)	Octet 1 bits 8 7 6 5
Call set-up packets	0 1 0 1
Clearing, flow control, interrupt, reset, restart, and diagnostic packets	0 0 0 1
Data packets	X X 0 1

Where:

a bit marked X may be either a 0 or a 1 as indicated elsewhere in the text.

Table 7/AX.25
General Format Identifier

6.1.2 Logical channel group number

The Logical channel group number (LCGN) is in all packets except for restart or diagnostic packets. It is binary encoded, and resides in bit positions 4, 3, 2, and 1 of octet 1, with bit 1 being the low order bit. For each logical channel, this number has local significance at the DTE/DCE interface.

In restart and diagnostic packets the logical channel group number is set to all zeros.

6.1.3 Logical channel number

The logical channel number (LCN) is in all packets except for restart or diagnostic packet!. It is binary encoded, and resides in all bit positions of octet 2, with bit 1 being the low order bit. For each logical channel, the LCN has local significance at the DTE/DCE interface.

In restart and diagnostic packets the logical channel number is set to all zeros.

6.1.4 Packet type identifier

Each packet type will be identified by the encoding of octet 3 of the packet. This identifier takes all bit positions, and is encoded as shown in Table 8/AX.25 (Table 8/AX.25 is at the end of this paper). Packet type identifiers other than those shown in Table 8/AX.25 are reserved.

6.2 Call set-up and clearing packets

6.2.1 Call request and incoming call packets

Figure 1/AX.25 illustrates the format of call request and incoming call packets.

```
                 Bit Position
          8   7   6   5        3   2   1
        ---------------------------------------
    1 ! General Format      ! Logical Channel !
      ! Identifier (GFI)    ! Group Number    !
 O    !  0  0  1  0  1      !    (LCGN)       !
 C    !-------------------------------------- !
 T  2 !   Logical Channel Number (LCN)        !
 E    !---------------------------------------!
 T    !        Packet Type Identifier         !
    3 !  0   0   0   0     1   0   1   1      !
      !---------------------------------------!
      ! Calling DTE         !  Called DTE     !
    4 ! Address Length      ! Address Length  !
      !---------------------------------------!
    V            DTE Addresses
    A !          (see Note 1)                 !
    R                   ----------------------
      !                 !  0   0   0   0      !
      !---------------------------------------!
      !   0   0 !   Facility Length           !
      !---------------------------------------!
    V
    A !             Facilities                !
    R
        --------------------------------------!
    V !             Call User Data            !
    A            (see Notes 2 and 3)
    R !                                       !
        ---------------------------------------
```

Note 1. The figure is drawn assuming the total amount of address information is not an integral number of octets. Each address sub-field is potentially of variable length, any padding necessary is added at the end, and will consist of zeros.

Note 2. Bits 8 and 7 of the first octet of the call user data field may have particluar significance (see 6.2.1).

Note 3. Maximum length of the call user data field is 16 octets.

Figure 1/AX.25
Call request and incoming packet format

6.2.1.1 General format identifier

Bit 8(Q bit) shall be set to 0, bit 7 (D bit) shall be set to 1, and bits 6 and 5 should be set to 0 and 1 respectively.

6.2.1.2 Address lengths field

Octet 4 consists of field length indicators for the called and calling DTE address fields. Bits 4, 3, 2, and 1 indicate the called DTE address length in semi-octets (nibbles). Bits 8, 7, 6, and 5 indicate the calling DTE address length in nibbles. Each address length indicator is encoded in binary, with bits 1 and 5 being the low order bits of their respective indicators.

6.2.1.3 Address field

Octet 5 and the following octets (up to 32 octets) consist of the called DTE address, followed by the calling DTE address. These addresses are encoded as described in Annex F.

6.2.1.4 Facility length field

The octet following the DTE address fields contains the facility length field. This octet has bits 8 and 7 unassigned, and both are set to 0. Bits 6, 5, 4, 3, 2, and 1 contain the facility length information. This information is binary coded, with bit 1 being the low order bit.

6.2.1.5 Facility field

The facility field is present only when the DTE is using an optional user facility requiring some indication in the call request and incoming call packets.

The coding of the facility field is described in section 7.

The facility field must contain an integral number of octets. The actual maximum length of this field depends on the facilities which are offered by the packet switch and network. The maximum number must not exceed 63 octets at any time.

6.2.1.6 Call user data field

A call user data field may be present following the facilities field. This field may be up to 16 octets long, and must contain an integral number of octets.

If a call user data field is present, the use and format of this field are determined by bits 8 and 7 of the first octet of this field in accordance with the following:

If bits 8 and 7 are 00, a portion of the call user data field is used for protocol identification in accordance with other Recommendations (such as AX.29).

If bits 8 and 7 are set to 01, a portion of the call user data field may be used for protocol identification in accordance with specifications of networks.

If bits 8 and 7 are 10, a portion of the call user data field may be used for protocol identification in accordance with specifications of international user bodies.

If bits 8 and 7 are set to 11, no constraints are placed on the use by the DTE of the remainder of the call user data field.

Users are cautioned that if bits 8 and 7 have a value other than 11, a protocol may be identified that is implemented within the network.

When a virtual call is established between two packet mode DTEs, the network does not act on any part of the call user data field.

6.2.2 Call accepted and call connected packets

Figure 2/AX.25 illustrates the format of call accepted and call connected packets.

```
                  Bit Position
           8   7   6   5     4   3   2   1
        ----------------------------------------
       !                   !                   !
    1  ! General Format    !  Logical Channel  !
       ! Identifier (GFI)  !  Group Number     !
 O     !  0   1   0   1    !     (LCGN)        !
 C     !---------------------------------------!
 T  2  !   Logical Channel Number (LCN)        !
 E     !---------------------------------------!
 T     !        Packet Type Identifier         !
    3  !  0   0   0   0      1   0   1   1     !
       !---------------------------------------!
       ! Calling DTE       !  Called DTE       !
    4  ! Address Length    !  Address Length   !
       !---------------------------------------!
    V             DTE Addresses
    A  !          (see Note 1)                 !
    R                   ---------------------
       !                   !  0   0   0   0    !
       !---------------------------------------!
       !   0   0 !   Facility Length           !
       !---------------------------------------!
    V
    A  !           Facilities                  !
    R             (see Note 2)
       !                                       !
        ---------------------------------------
```

Note 1. The figure is drawn assuming the total amount of address information is not an integral number of octets. Each address sub-field is potentially of variable length, any padding necessary is added at the end, and will consist of zeros.

Note 2. The facility field is not mandatory in call accepted packet (see 6.2.2).

Figure 2/AX.25
Call accepted and call connected packet format

6.2.2.1 General format identifier

Bit 8(Q bit) shall be set to 0, bit 7 (D bit) shall be set to 1, and bits 6 and 5 should be set to 0 and 1 respectively.

6.2.2.2 Address lengths field

Octet 4 consists of field length indicators for the called and calling DTE address fields. Bits 4, 3, 2, and 1 indicate the called DTE

address length in semi-octets (nibbles). Bits 8, 7, 6, and 5 indicate the calling DTE address length in nibbles. Each address length indicator is encoded in binary, with bits 1 and 5 being the low order bits of their respective indicators.

6.2.2.3 Address field

Octet 5 and the following octets (up to 32 octets) consist of the called DTE address, followed by the calling DTE address. These addresses are encoded as described in Annex F.

6.2.2.4 Facility length field

The octet following the DTE address fields contains the facility length field. This octet has bits 8 and 7 unassigned, and both are set to 0. Bits 6 5, 4, 3, 2, and 1 contain the facility length information. This information is binary coded, with bit 1 being the low order bit.

The use of the facility length field in call accepted packets is mandatory. It should be set to all zeros if there is no facility field.

6.2.2.5 Facility field

The facility field is present only when the DTE is using an optional user facility requiring some indication in the call accepted and call connected packets.

The coding of the facility field is described in section 7.

The facility field must contain an integral number of octets. The actual maximum length of this field depends on the facilities which are offered by the packet switch and network. The maximum number must not exceed 63 octets at any time.

6.2.3 Clear request and clear indication packets

Figure 3/AX.25 shows the format for clear request and clear indication packets.

```
               8  7  6  5 Bit Position 4  3  2  1
             ---------------------------------------
             ! General Format      ! Logical Channel !
  O    1     ! Identifier (GFI) 0  ! Group Number    !
  C          !--------------------------------------!
  T    2     !   Logical Channel Number (LCN)        !
  E          !--------------------------------------!
  T          !          Packet Type                  !
       3     !   0    0    0 1    Identifier    1 1   !
                            0    0
       4     !           Clearing Cause              !
             ---------------------------------------
       5     !           Diagnostic Code             !
             ---------------------------------------
```

Figure 3/AX.25.
Clear request and clear indication packet format

6.2.3.1 Clearing cause field

Octet 4 is the clearing cause field, which contains the reason for the clearing of the call.

In clear request packets, the clearing cause field should be set by the DTE to one of the following values:

```
bit:     8 7 6 5 4 3 2 1
value1:  0 0 0 0 0 0 0 0
value2:  1 X X X X X X X
where X may be either 0 or 1
```

The DCE will prevent values other than those specified above in the clearing cause field from reaching the other end of the call by considering the clear request as an error and following the procedure described in Annex C.

The coding of the clearing cause field in clear indication packets is given in Table 9/AX.25 (Table 9/AX.25 is at the end of this paper).

6.2.3.2 Diagnostic code

Octet 5 is the diagnostic code which contains additional information on the reason for the clearing of the call.

In a clear request packet, the diagnostic code is not mandatory.

In a clear indication packet, if the clearing cause field indicates "DTE originated", the diagnostic code is passed unchanged from the clearing DTE. If the clearing DTE has not provided a diagnostic code in its clear request packet, then the bits of the diagnostic code in the resulting clear indication packet will all be zero.

When a clear indication packet results from a restart request packet, the value of the diagnostic code will be that specified in the restart request packet, or all zeros in the case where no diagnostic code has been specified in restart request.

When the clearing cause field does not indicate "DTE originated", the diagnostic code in a clear indication packet is network generated. Annex E lists the codings for network generated diagnostics. The bits of the diagnostic code are all set to zero when no specified additional information for the clearing is supplied.

The contents of the diagnostic code field do not alter the meaning of the cause field. A DTE is not required to undertake any action on the contents of the diagnostic code field. Unspecified code combinations in the diagnostic code field shall not cause the DTE to refuse the cause field.

6.2.4 DTE and DCE clear confirmation packets

Figure 4/AX.25 shows the format of the DTE and DCE clear indication packets.

```
                        Bit
            8  7  6  5  Position  4  3  2  1
          ------------------------------------
       1  ! General Format   ! Logical Channel !
  O       !  Identifier (GFI) ! Group Number   !
  C       !----------------------------------!
  T    2  !  Logical Channel Number (LCN)     !
  E       !----------------------------------!
  T    3  !  0  0  Packet  Type  Identifier 1 1 !
          -----------0------1------0----1-------
```

Figure 4/AX.25
DTE and DCE clear confirmation packet format

6.3 Data and interrupt packets

6.3.1 DTE and DCE data packets

Figure 5/AX.25 illustrates the format of the DTE and DCE data packets.

```
                  Bit Position
            8  7  6  5     4  3  2  1
          ------------------------------------
          ! General Format  ! Logical Channel !
       1  ! Identifier (GFI) ! Group Number   !
  O       !  Q  D  0  1     !                 !
  C       !----------------------------------!
  T    2  !  Logical Channel Number (LCN)     !
  E       !----------------------------------!
  T    3  !  P(R)    !M  !    P(S)     ! 0    !
          !----------------------------------!
  V       !                                  !
  A       !            User Data             !
  R       !                                  !
          ------------------------------------
```

Where: D is the Delivery confirmation bit
M is the More data bit
Q is the data Qualifier bit

Figure 5/AX.25. DTE and DCE data packet format

6.3.1.1 Qualifier (Q) bit

Bit 8 of octet 1 is the qualifier bit (Q bit). Q bit operation is described in section 4.3.6.

6.3.1.2 Delivery confirmation (D) bit

Bit 7 of octet 1 is the delivery confirmation bit (D bit). D bit operation is described in section 4.4.1.4.

6.3.1.3 Packet receive sequence number

Bits 8, 7, and 6 of octet 3 are used for indicating the packet receive sequence number P(R). P(R) is binary coded with bit 6 being the low order bit.

6.3.1.4 More data bit

Bit 5 in octet 3 is used for the More data mark (M bit); 0 for no more data, 1 for more data,

6.3.1.5 Packet send sequence number

Bits 4, 3, and 2 of octet 3 are used for indicating the packet send sequence number P(S). P(S) is binary coded, with bit 2 being the low order bit.

6.3.1.6 User data field

Octets following the third octet contain user data. The user data field must contain an integral number of octets.

6.3.2 DTE and DCE interrupt packets

Figure 6/AX.25 illustrates the format of the DTE and DCE interrupt packets.

```
                    Bit Position
         8   7   6   5               3   2   1
        ------------------------------------------
        ! General Format    ! Logical Channel !
      1 ! Identifier (GFI)  ! Group Number    !
 O      !  0   0   0   1    !                 !
 C      !-------------------  - - - - - - - - !
 T    2 !    Logical Channel Number (LCN)     !
 E      !-------------------------------------!
 T      !        Packet Type Identifier       !
      3 !  0   0   10       0   0   1   1     !
        !-------------------------------------!
      4 !          Interrupt User Data        !
        ---------------------------------------
```

Figure 6/AX.25.
DTE and DCE interrupt packet format

6.3.2.1 Interrupt user data field

Octet 4 contains user data.

6.3.3 DTE and DCE Interrupt confirmation packets

Figure 7/AX.25 illustrates the format of the DTE and DCE interrupt confirmation packets.

```
                    Bit Position
         8   7   6   5       4   3   2   1
        ------------------------------------------
        ! General Format    ! Logical Channel !
      1 ! Identifier (GFI)  ! Group Number    !
 O      !  0   0   0   1    !                 !
 C      !-------------------  - - - - - - - - !
 T    2 !    Logical Channel Number (LCN)     !
 E      !-------------------------------------!
 T      !        Packet Type Identifier       !
      3 !  0   0   10       0   1   1   1     !
        ---------------------------------------
```

Figure 7/AX.25
DTE and DCE interrupt confirmation packet format

6.4 Datagram and datagram service signal packets

Datagrams are not implemented in AX.25.

6.5 Flow control and reset packets

6.5.1 DTE and DCE receive ready (RR) packets

Figure 10/AX.25 shows the format of the DTE and DCE receive ready (RR) packets.

```
         8   7   6   Bit Position   3   2   1
                   5        4
        ------------------------------------------
        ! General Format    ! Logical Channel !
      1 ! Identifier (GFI)  ! Group Number    !
 O      !  0   0   0   1    ! 0               !
 C      !-------------------  - - - - - - - - !
 T    2 !    Logical Channel Number (LCN)     !
 E      !-------------------------------------!
 T      !             ! Packet Type Identifier!
      3 !   P(R)      !  0    0    0    0  1  !
        ---------------------------------------
```

Pigure 10/AX.25
DTE and DCE RR packet format

6.5.1.1 Packet receive sequence number

Bits 8, 7, and 6 of octet 3 are used for indicating the packet receive sequence number P(R). P(R) is binary coded, with bit 6 being the low order bit.

6.5.2 DTE and DCE receive not ready (RNR) packets

Figure 11/AX.25 illustrates the format of the DTE and DCE RNR packets.

```
                    Bit Position
         8   7   6   5       4   3   2   1
        ------------------------------------------
      1 ! General Format    ! Logical Channel !
        ! Identifier (GFI)  ! Group Number    !
 O      !  0   0   0   1    !                 !
 C      !-------------------  - - - - - - - - !
 T    2 !    Logical Channel Number (LCN)     !
 E      !-------------------------------------!
 T      !             ! Packet Type Identifier!
      3 !   P(R)      ! 0     0    10     1   !
        ---------------------------------------
```

Figure 11/AX.25
DTE and DCE RNR packet formats

6.5.2.1 Packet receive sequence number

Bits 8, 7, and 6 of octet 3 are used for indicating the packet receive sequence number P(R). P(R) is binary coded, with bit 6 being the low order bit.

6.5.3 Reset request and reset indication packets

Figure 12/AX.25 illustrates the format of the reset request and reset indication packets.

```
                    Bit Position
         8   7   6   5       4   3   2   1
        ------------------------------------------
        ! General Format    ! Logical Channel !
      1 ! Identifier (GFI)  ! Group Number    !
 O      !  0   0   0   1    !   (LCGN)        !
 C      !-------------------  - - - - - - - - !
 T    2 !    Logical Channel Number (LCN)     !
 E      !-------------------------------------!
 T      !        Packet Type Identifier       !
      3 !  0   0   0   1     1   0   1   1    !
        !-------------------------------------!
      4 !           Resetting Cause           !
        !-------------------------------------!
      5 !           Diagnostic Code           !
        ---------------------------------------
```

Figure 12/AX.25
Reset request and reset indication packet format

6.5.3.1 Resetting cause field

Octet 4 is the resetting cause field, which contains the reason for the reset.

In reset request packets, the resetting cause field should be set by the DTE to one of the following values:

bit:	8	7	6	5	4	3	2	1
value1:	0	0	0	0	0	0	0	0
value2:	1	X	X	X	X	X	X	X

where X may be either 0 or 1

The DCE will prevent values other than those specified above in the resetting cause field from reaching the other end of the call by considering

the reset request as an error and following the procedure described in Annex C.

The coding of the resetting cause field in the reset indication packets is given in Table 11/AX.25 (Table 11/AX.25 is at the end of this paper).

6.5.3.2 Diagnostic code

Octet 5 is the diagnostic code which contains additional information on the reason for the reset.

In a reset indication packet, if the resetting cause field indicates "DTE originated", the diagnostic code is passed unchanged from the resetting DTE. If the DTE requesting a reset has not provided a diagnostic code in its reset request packet, then the bits of the diagnostic code in the resulting reset indication packet will all be zero.

When a reset indication packet results from a restart request packet, the value of the diagnostic code will be that specified in the restart request packet, or all zeros in the case where there is no diagnostic code has been specified in restart request packet.

When the resetting cause field does not indicate "DTE originated", the diagnostic code in a reset indication packet is network generated. Annex E lists the codings for network generated diagnostics. The bits of the diagnostic code are all set to zero when no specified additional information for the reset is supplied.

The contents of the diagnostic code field do not alter the meaning of the cause field. A DTE is not required to undertake any action on the contents of the diagnostic code field. Unspecified code combinations in the diagnostic code field shall not cause the DTE to refuse the cause field.

6.5.4 DTE and DCE reset confirmation packets

Figure 13/AX.25 shows the format of the DTE and DCE reset confirmation packets.

Bit Position

OCTET	8	7	6	5	4	3	2	1
1	General Format Identifier (GFI)				Logical Channel Group number (LCGN)			
2	Logical Channel Number (LCN)							
3	Packet Type Identifier							
	0	0			1	1	1	1

Figure 13/AX.25
DTE and DCE reset confirmation packets

6.6.1 Restart request and restart indication Packets

Figure 14/AX.25 illustrates the format of the restart request and restart indication packets.

Bit Position

OCTET	8	7	6	5	4	3	2	1
1	General Format Identifier (GFI)							
	0	0	0	1	0	0	0	0
2	0	0	0	0	0	0	0	0
3	Packet Type Identifier							
	1	1			1	0	1	1
4	Restarting Cause							
5	Diagnostic Code							

Figure 14/AX.25
Restart request and restart indication packet format

6.6.1.1 Restarting cause field

Octet 4 is the restarting cause field, which contains the reason for the restart.

In restart request packets the restarting cause field should be set by the DTE to one of the following values:

bitt	8	7	6	5	4	3	2	1
valuel:	0	0	0	0	0	0	0	0
value2 :	1	X	X	X	X	X	X	X

where X may be either 0 or 1

The DCE will prevent values other than those specified above in the restarting cause field from reaching the other end of the call by considering the restart request as an error and following the procedure described in Annex C.

The coding of the restarting cause field in the restart indication packets is given in Table 12/AX.25.

	8	7	6	5	4	3	2	1
Local procedure error	0	0	0	0	0	0	0	1
Network congestion	0	0	0	0	0	0	1	1
Network operational	0	0	0	0	0	1	1	1

Table 12/AX.25
Coding of the restarting cause field in restart indication packets

6.6.1.2 Diagnostic code

Octet 5 is the diagnostic code which contains additional information on the reason for the restart.

The diagnostic code is passed to the corresponding DTEs as the diagnostic code of a clear indication packet for virtual calls.

The coding in a restart indication packet is given in Annex E. The bits of the diagnostic code are all set to zero when no specified additional information for the restart is supplied.

The contents of the diagnostic code field do not alter the meaning of the cause field. A DTE is not required to undertake any action on the contents of the diagnostic code field. Unspecified code combinations in the diagnostic code field shall not cause the DTE to refuse the cause field.

6.6.2 DTE and DCE restart confirmation packets

Bit Position

OCTET	8	7	6	5	4	3	2	1
1	General Format Identifier (GFI)							
	0	0	0	1	0	0	0	0
2	0	0	0	0	0	0	0	0
3	Packet Type Identifier							
	1	1			1	1	1	1

Figure 15/AX.25
DTE and DCE restart confirmation packet format

6.7 Diagnostic packet

Figure 16/AX.25 shows the format of the diagnostic packet.

Bit Position

OCTET	8	7	6	5	4	3	2	1
1	General Format Identifier (GFI)				0	0	0	0
	0	0	0	1				
2	0	0	0	0	0	0	0	0
3	Packet Type Identifier							
	1	1			0	0	0	1
4	Diagnostic Code							
5	Diagnostic Explanation (see Note 1)							

Note 1: The figure drawn assumes an integral number of octets in the diagnostic explanation

Figure 16/AX.25 Diagnostic packet format

6.7.1 Diagnostic code field

Octet 4 is the diagnostic code and contains information on the error condition that caused the transmission of the diagnostic packet. The coding of the diagnostic field is given in Annex E.

6.7.2 Diagnostic explanation field

When the diagnostic packet is issued as a result of the reception of an erroneous packet from the DTE (see Table C-1/AX.25), this field contains the first three octets of header information from the erroneous DTE packet. If the packet contains less than 3 octets, this field contains whatever bits were received.

When the diagnostic packet is issued as a result of a DCE time-out (see Table D-1/AX.25), the diagnostic explanation field contains 2 octets coded as follows:

bits 8, 7, 6, and 5 of the first octet contain the general format identifier for the interface;

bits 4 to 1 of the first octet and bits 8 to 1 of the second octet are all zero for the expiration of time-out T10 and give the number of the logical channel on which the time-out occured for expiration of time-out T12 or T13.

6.8.2 Set-up and clearing packets for the fast select facility and fast select acceptance facility

6.8.2.1 Call request and incoming call packets

Figure 18/AX.25 illustrates the format of call request and incoming call packets used in conjunction with the fast select facility described in section 7.2.4.

The description in section 6.2.1 applies here, except that the length of the call user data field has a maximum length of 128 octets, and should contain an integral number of octets.

```
                    Bit Position
            8   7   6   5       4   3   2   1
          ---------------------------------------
     1  ! General Format    ! Logical Channel  !
        ! Identifier (GFI)  ! Group Number     !
O       !  0   1   0   1    !    (LCGN)        !
C       !------------------ - - - - - - - - - -!
T    2  !    Logical Channel Number (LCN)      !
E       !--------------------------------------!
T       !        Packet Type Identifier        !
     3  !  0   0   0   0      1   0   1 1      !
        !--------------------------------------!
        ! Calling DTE       !  Called DTE      !
     4  ! Address Length    ! Address Length   !
        !--------------------------------------!
     V          DTE Addresses
     A !        (see Note 1)                   !
     R                    ----------I---------
        !                   !  0   0   0   0   !
        !--------------------------------------!
        !   0   0 !   Facility Length          !
        !--------------------------------------!
     V
     A !             Facilities                !
     R
        !--------------------------------------!
     V I            Call User Data             !
     A           (see Notes 2 and 3)
     R !                                       !
          ---------------------------------------
```

Note 1. The figure is drawn assuming the total amount of address information is not an integral number of octets. Each address sub-field is potentially of variable length, any padding necessary is added at the end, and will consist of zeros.

Note 2. Bits 8 and 7 of the first octet of the call user data field ma have particluar significance (see 6.2.1 3

Note 3. Maximum length of the call user data field is 128 octets.

Figure **18/AX.25**
Call request and incoming call packet format for the fast select facility

6.8.2.2 Call accepted and call connected packets

Figure 19/AX.25 illustrates the format of call accepted and call connected packets used in conjunction with the fast select facility described in section 7.2.4.

```
                    Bit Position
            8   7   6   5       4   3   2   1
          ---------------------------------------
        ! General Format    !  Logical Channel !
     1  ! Identifier (GFI)  ! Group Number     !
0       !  00  1  0  1      !    (LCGN)        !
C       !------------------ - - - - - - - - - -!
T    2  !    Logical Channel Number (LCN)      !
E       !--------------------------------------!
T       !        Packet Type Identifier        !
     3  !  0   0   0   0      1   1   1   1    !
        !--------------------------------------!
        ! Calling DTE       !  Called DTE      !
     4  ! Address Length    ! Address Length   !
        !--------------------------------------!
     V           DTE Addresses
     A !         (see Note 1)                  !
     R                   -------------------
        !                   !  0   0   0   0   !
        !--------------------------------------!
        !   0   0 !   Facility Length          !
        !--------------------------------------!
     V
     A !             Facilities                !
     R
        !                                      !
        !--------------------------------------!
        !                                      !
                      Call User Data
        !           (See Notes 2 and 3)        !
          ---------------------------------------
```

Note 1. The figure is drawn assuming the total amount of address information is not an integral number of octets. Each address sub-field is potentially of variable length, any padding necessary is added at the end, and will consist of zeros.

Note 2. Bits 8 and 7 of the first octet of the call user data field may have particluar significance (see 6.8.2.2).

Note 3. Maximum length of the call user data field is 128 octets.

Figure **19/AX.25**
Call **accep**ted and **call** connected packet format for the fast select facilitp

The description in section 6.2.2 applies here. In addition, a call user data field may be present. If a call user data field is present, it can contain up to a maximum of 128 octets, and must contain an integral number of octets.

If a call user data field is present the use and format of this field are determined by bits 8 and 7 of the first octet of this field in accordance with the following:

If bits 8 and 7 are 00, a portion of the call user data field is used for protocol identification in accordance with other Recommendations (such as AX.29).

If bits 8 and 7 are set to 01, a portion of the call user data field may be used for protocol identification in accordance with specifications of networks.

If bits 8 and 7 are 10, a portion of the call user data field may be used for protocol identification in accordance with specifications of international user bodies.

If bits 8 and 7 are set to 11, no constraints are placed on the use by the DTE of the remainder of the call user data field.

Users are cautioned that if bits 8 and 7 have a value other than 11, a protocol may be identified that is implemented within the network.

When a virtual call is established between two packet mode DTEs, the network does not act on any part of the call user data field.

6.8.2.3 Clear request and clear indication packets

Figure 20/AX.25 illustrates the format of clear request and clear indication packets used in conjunction with the fast select facility and fast select acceptance facility described in sections 7.2.4 and 7.2.5.

```
                 Bit Position
        8   7   6   5       4   3   2   1
       ----------------------------------------
       ! General Format   !  Logical Channel !
     1 ! Identifier (GFI) !  Group Number    !
 O     !  0   0   0   1   !    (LCGN)        !
 C     !------------------ ----------------- !
 T   2 !   Logical Channel Number (LCN)      !
 E     !-------------------------------------!
 T     !      Packet Type Identifier         !
     3 !  0   0   0   1       0   0   11     !
       !-------------------------------------!
     4 !          Clearing Cause             !
       !-------------------------------------!
     5 !          Diagnostic     Code        !
       !-------------------------------------!
       ! Calling DTE      !  Called DTE      !
     6 ! Address Length   !  Address Length  !
                                             !
 V               DTE Addresses
 A !            (see Note 1)
 R                  -----I------------m-     !
       !                  !  0   0   0   0   !
       !-------------------------------------!
       !   0   0   !   Facility   Length     !
       !-------------------------------------!
 V
 A !               Facilities                !
 R
       !                                     !
       !-------------------------------------!
       !                                     !
                   Call User Data
       !           (See Note 2)              !
       -*-----I-------------------------------
```

Note 1. The figure is drawn assuming the total amount of address information is not an integral number of octets. Each address sub-field is potentially of variable length, any padding necessary is added at the end, and will consist of zeros.

Note 2. Maximum length of the call user data field is 128 octets.

Figure 20/AX.25
Clear request and clear indication packet format for the fast select facility and clear indication packet format to the calling DTE for the called called line address modified notification facility

The descriptions of the clearing cause field and the diagnostic code field in section 6.2.3 apply here. In additiion, the following fields may follow the diagnostic code field (the use of the diagnostic field itself is mandatory).

6.8.2.3.1 Address length field

This field is coded with all zeros.

6.8.2.3.2 Address field

This field is not present.

6.8.2.3.3 Facility length field

This field is coded with all zeros.

6.8.2.3.4 Facility field

This field is not present.

6.8.2.3.5 Clear user data field

Following the facility field, a clear user data field may be present, and if present, it may be up to a maximum of 128 octets. The clear user data field is required to contain an integral number of octets.

6.8.4 Clear indication packet for called line address modified notification facility

Figure 20/AX.25 illustrates the format of clear indication packet used in conjunction with the called line address modified notification facility described in section 7.2.10.

The description in section 6.8.2.3 applies here, except the notes attached to sections 6.8.2.3.1 and 6.8.2.3.3 and all of sections 6.8.2.3.2, 6.8.2.3.4, and 6.8.2.3.5.

Address field

The addess field is present only when the call is redirected and then cleared by the network or by the alternate DTE without the transmission of the call accepted packet. It consists of the alternate DTE address.

The coding of the address field is described in section 6.2.1.3.

Facility field

The facility field is present only when the call is redirected and then cleared by the network or by the alternate DTE without the transmission of the call accepted packet.

The coding of the facility field is defined in section 7.

From DCE to DTE	From DTE to DCE	Octet 3 Bits 8 7 6 5 4 3 2 1
Call set-up and clearing		
Incoming call	Call request	0 0 0 0 1 0 1 1
Call connected	Call accepted	0 0 0 0 1 1 1 1
Clear indication	Clear request	0 0 0 1 0 0 1 1
DCE clear confirmation	DTE clear confirmation	0 0 0 1 0 1 1 1
Data and interrupt		
DCE data	DTE data	x x x x x x x 0
DCE interrupt	DTE interrupt	0 0 1 0 0 0 1 1
DCE interrupt	DTE interrupt confirmation	0 0 0 1 0 1 1 1
Flow control and reset		
DCE RR	DTE RR	x x x 0 0 0 0 1
DCE RNR	DTE RNR	x x x 0 0 1 0 1
Reset indication	Reset request	0 0 0 1 1 0 1 1
DCE reset confirmation	DTE reset confirmation	0 0 0 1 1 1 1 1
Restart		
Restart indication	Restart request	1 1 1 1 1 0 1 1
DCE restart confirmation	DTE restart confirmation	1 1 1 1 1 1 1 1
Diagnostic		
Diagnostic		1 1 1 1 0 0 0 1

Note: an x bit may be either a 0 or 1

Table 8/AX.25. Packet type identifier encodings

	Bits 8 7 6 5 4 3 2 1
DTE originated	0 0 0 0 0 0 0 0
DTE originated (b)	1 x x x x x x x
Number busy	0 0 0 0 0 0 0 1
Out of order	0 0 0 0 1 0 0 1
Remote procedure error	0 0 0 1 0 0 0 1
Reverse charging acceptance not subscribed (d)	0 0 0 1 1 0 0 1
Incompatible destination	0 0 1 0 0 0 0 1
Fast select acceptance not subscribed	0 0 1 0 1 0 0 1
Destination absent	0 0 1 1 1 0 0 1
Invalid facility request	0 0 0 0 0 0 1 1
Access barred	0 0 0 0 1 0 1 1
Local procedure error	0 0 0 1 0 0 1 1
Network congestion	0 0 0 0 0 1 0 1
Not obtainable	0 0 0 0 1 1 0 1
RPOA out-of-order (a)	0 0 0 1 0 1 0 1

Where :

(a) May be received only if the corresponding optional user facility is used.

(b) When bit 8 is set to 1, the bits represented by Xs are those included by the remote DTE in the clearing or restarting cause field of the clear or restart request packet.

(d) Used for amateur to public data network internetworking only.

Table 9/AX.25

Coding of clearing cause field in clear indication packet

	Bits 8 7 6 5 4 3 2 1
DTE originated	0 0 0 0 0 0 0 0
DTE originated (d)	1 x x x x x x x
Remote procedure error	0 0 0 0 0 0 1 1
Incompatible destination	0 0 0 1 0 0 0 1
Local procedure error	0 0 0 0 0 1 0 1
Network congestion	0 0 0 0 0 1 1 1

Where:

(d) When bit 8 is set to 1, the bits represented by Xs are those indicated by the remote DTE in the resetting cause field of the reset request packet.

Table 11/AX.25

Coding of resetting cause field in reset indication packet

OPTIONAL FACILITIES FOR AX.25 LEVEL 3 PROTOCOL

Terry Fox, WB4JFI
President, AMRAD
1819 Anderson Rd.
Falls Church, VA 22043

Description

This paper is the third in a series of papers that make up a recommendation for the AX.25 Network Sublayer protocol.

The purpose of this paper is to describe optional user facilities requested of the network or called DTE at time of a call request. Included in these facilities are standard CCITT recommended facilities, and additional amateur network facilities. The amateur facilities are suggested by the draft committee, reader suggestions or comments are invited.

Comments may be sent to the author, or sent to the AMRAD Newsletter for publication. Follow the AMRAD Newsletter for further details.

7 Procedures and formats for optional user facilities ate packet level

7.1 Procedures for optional user facilities associated with virtual call service

7.1.1 Extended packet sequence numbering

This facility is not presently allowed in AX.25.

7.1.2 Nonstandard default window sizes

Nonstandard default window sizes is an optional user facility agreed to for a period of time. This facility, if subscribed to, provides for the selection of default window sizes from the list of window sizes supported. Some networks may constrain the default window sizes to be the same for each direction of data transmission across the DTE/DCE interface. In the absence of this facility, the default window sizes are 2.

Values other than the default window sizes may be negotiated for a virtual call by means of the flow control parameter negotiation facility (see 7.2.2).

7.1.3 Default throughput classes assignment

This optional facility is not implemented in AX.25.

7.1.4 Packet retransmission

This optional facility is not implemented in AX.25.

7.1.5 Incoming calls barred

This optional facility is not implemented in AX.25.

7.1.6 Outgoing calls barred

This optional facility is not implemented in AX.25.

7.1.7 One-way logical channel outgoing

One-way logical channel outgoing is an optional user facility agreed to for a period of time. This user facility, if subscribed to, restricts the logical channel use to originating outgoing virtual calls only.

A logical channel used for virtual calls retains its full duplex capability.

The rules according to which logical channel group numbers and logical channel numbers can be assigned to one-way outgoing logical channels for virtual calls are given in Annex A.

If all the logical channels for virtual calls are one-way outgoing at a DTE/DCE interface, the effect is equivalent to the incoming calls barred facility (not implemented).

7.1.8 One-way logical channel incoming

One-way logical channel incoming is an optional user facility agreed to for a period of time. This user facility, if it is supported, restricts the logical channel use to receiving incoming virtual calls only. A logical channel used for virtual calls retains its full duplex capability.

The rules according to which logical channel group numbers and logical channel numbers can be assigned to one-way incoming logical channels for virtual calls are given in Annex A.

If all the logical channels for virtual calls are one-way incoming at a DTE/DCE interface, the effect is equivalent to the outgoing calls barred facility (not implemented).

7.1.9 Closed user group

This optional facility is not supported.

7.1.10 Closed user group with outgoing access

This optional facility is not available in AX.25.

7.1.11 Closed user group with incoming access

This optional facility is not available in AX.25.

7.1.12 Incoming calls barred within a closed user group

This optional facility is not available in AX.25.

7.1.13 Outgoing calls barred within a closed user group

This optional facility is not available in AX.25.

7.1.14 Bilateral closed user group

This optional facility is not available in AX.25.

7.1.15 Bilateral closed user group with outgoing ac cess ---

This optional facility is not available in AX.25.

7.1.16 Reverse charging

Reverse charging is an optional user facility which mav be requested by a DTE for a given virtual call (see 7.4.2.3), and only for the case of a virtual call destined for a DTE on a public data network.

7.1.17 Reverse charging acceptance

This optional facility is not available in AX.25.

7.1.18 RPOA selection

Recognized private operating agency (RPOA) selection is an optional user facility which may be requested by a DTE for a given virtual call.

When this user facility is requested, it provides for the user specification by the calling/source DTE of a particular RPOA transit network through which the call is to be routed internationally, when more than one RPOA transit network exists at a gateway (see 7.4.2.4).

7.2 Procedures for optional user facilities only available with virtual call services

7.2.1 Nonstandard default packet sizes

Nonstandard default packet sizes is an optional user facility agreed to for a period of time. This facility, if subscribed to, provides for the selection of default packet sizes from the list of packet sizes supported. Some networks may constrain the packet sizes to be the same for each direction of data transmission across the DTE/DCE interface. In the absence of this facility, the default packet sizes are 128 octets. The term "packet sizes" refers to the maximum user data field lengths of DCE and DTE data packets.

Values other than the default packet sizes may be negotiated for a virtual call by means of the flow control parameter negotiation facility (see 7.2.2).

7.2.2 Flow control parameter negotiation

Flow parameter negotiation is an optional user facility agreed to for a period of time which can be used by a DTE on virtual calls. This facility, if subscribed to, permits negotiation on a per-call basis of the flow control parameters. The flow control parameters considered are the packet and window sizes at the DTE/DCE interface for each direction of data transmission.

"Packet sizes" in 7.2.2 refers to the maximum user data field lengths of DCE and DTE data packets.

In the absence of the flow control parameter negotiation facility, the flow control parameters to be used at a particular DTE/DCE interface are the default packet sizes (see 7.2.1) and the default window sizes (see 7.1.2).

When the calling DTE has subscribed to the flow control parameter negotiation facility, it may separately request packet sizes and window sizes for each direction of data transmission (see 7.4.2.5). If a particular window size is not explicitly requested in a call request packet, the DCE will assume that the default window size was requested. If a particular packet size is not explicitly requested, the DCE will assume that the default packet size was requested.

When a called DTE has subscribed to the flow control parameter negotiation facility, each incoming call packet will indicate the packet and window sizes from which DTE negotiation can start. No relationship needs to exist between the packet sizes (P) and window sizes (W) requested in the call request packet and those indicated in the incoming call packet. The called DTE may request window and packet sizes with facilities in the call accepted packet. The only valid facility requests in the call accepted packet, as a function of the facility indications in the incoming call packet, are given in Table 13/AX.25 (Table 13/AX.25 is at the end of this paper). If the facility request is not made in the call accepted packet, the DTE is assumed to have accepted the indicated values (regardless of the default values).

When the calling DTE has subscribed to the flow control negotiation facility, every call connected packet will indicate the packet and window sizes to be used at the DTE/DCE interface for the call. The only valid facility indications in the call request packet are given in Table 14/AX.25 (Table 14/AX.25 is at the end of this paper).

The network may have constraints requiring the flow control parameters used for a call to be modified before indicating them to the DTE in the incoming call packet or call connected packet; the ranges of parameter values available on various networks may differ.

Window and packet sizes need not be the same at each end of a virtual call.

The role of the DCE in negotiating the flow control parameters may be network dependant.

7.2.3 Throughput class negotiation

This parameter is not presently implemented in AX.25.

7.2.4 Fast select

Fast select is an optional user facility which may be requested by a DTE for a given virtual call.

DTEs can request the fast select facility on a per-call basis by means of an appropriate facility request (see 7.4.2.7) in a call request packet using any logical channel which has been assigned to virtual calls.

The fast select facility, if requested in the call request packet and if it indicates no restriction on response, allows this packet to contain a call user data field of up to 128 octets and authorizes the DCE to transmit to the DTE, during the DTE waiting state, a call connected packet with a called user data field of up to 128 octets or a clear indication packet with a clear user data field of up to 128 octets.

The fast select facility, if requested in the call request packet and if it indicates restriction on response, allows this packet to contain a call user data field of up to 128 octets and authorizes the DCE to send to the DTE, during the DTE waiting state, a clear indication packet with a clear user data field of up to 128 octets; the DCE would not be authorized to transmit a call connected packet.

The presence of the fast select facility indicating no restriction on response in an incoming call packet permits the DTE to issue as a direct response to this packet a call accepted packet with a called user data field of up to 128 octets or a clear request packet with a clear user data field of up to 128 octets.

The presence of the fast select facility indicating restriction on response in an incoming call packet permits the DTE to issue as a direct response to this packet a clear request packet with a clear user data field of up to 128 octets; the DTE would not be authorized to send a call accepted packet.

A clear request packet with a clear user data field of up to 128 octets at any time other than the DCE waiting state (p3) is not allowed.

The call user data field, the called user data field, and the clear user data field will not be fragmented for delivery across the DTE/DCE interface.

The significance of the call connected packet and the clear indication packet with the cause "DTE originated" as a direct response to the call request packet with the fast select facility is that the call request packet with the data field has been received by the called DTE.

All other procedures of a call in which the fast select facility has been requested are the same as those of a virtual call.

If a fast select clear re uest packet with a non-zero address length fiel8 or facility length field is received, the DCE shall discard the received packet. The DCE shall indicate clearing by sending to the DTE a clear indication packet with a cause "Local procedure error" and diagnostic #74 or #75, as appropriate. The distant DTE is also inf'ormed of the clearing by a clear indication packet, with the cause "Remote procedure error" (same diagnostic).

7.2.5 Fast select acceptance

This optional facility is not implemented in AX.25. All users should allow fast select.

7.2.6 Charging information

This optional facility is not implemented in AX.25.

7.2.7 D bit modification

This optional facility is not implemented in AX.25.

7.2.8 Hunt group

Hunt group is an optional user facility agreed to for a period of time. This user facility, if subscribed to, distributes incoming calls having an address associated with the hunt group across a designated grouping of DTE/DCE interfaces.

Selection is performed for an incoming virtual call if there is at least one idle logical channel available for virtual calls on any DTE/DCE interface in a group. Once a virtual call is assigned to a DTE/DCE interface, it is treated as a regular call.

When virtual calls are placed to a hunt group address in the case specific addresses have also been assigned to the individual DTE/DCE interfaces, the call connected or clear indication packet sent to the calling DTE will optionally contain the called address of the selected DTE/DCE interface and the called line address modified notification facility indicating the reason why the called address is different from the one originally requested.

Virtual calls may be originated on DTE/DCE interfaces belonging to the hunt group; these are handled in the normal manner. In particular, the calling DTE address transferred to the remote DTE in the incoming call packet is the hunt group unless the DTE/DCE interface has a specific address assigned. Some networks may place a limit on the number of DTE/DCE interfaces in the hunt group, and/or constrain the size of the geographic region that can be served by a single hunt group.

7.2.9 Call redirection

Call redirection is an optional facility agreed to for a period of time. This user facility, if subscribed to by a DTE, redirects incoming calls destined to this DTE, when:

1) the called DTE is out-of-order, or

2) the called DTE is busy.

Some networks may provide call redirection only in the case of 1) above.

In addition, some networks may offer:

3) systematic call redirection with prior request of the called DTE.

The basic service is limited to one call redirection. In addition, some networks may offer either one of the following (mutually exclusive) capabilities:

A) A list of alternate DTEs (cl, c2, c3, . ..etc) is stored by the network of the originally called DTE (DTE B). consecutive attempts of call redirection are tried to each of these addresses, in the order of the list, up to the completion of the call;

B) Call redirections may be logically chained; if DTE C has subscribed to call redirection to DTE D, the call may be redirected to D even if it was originally addressed to B.

In any case, networks will ensure that loops are avoided and that connection establishment phase has a limited duration.

If a call is cleared by the network as a consequence of the actions taken to this effect, the clearing cause is the one generated at the last DTE/DCE interface.

When the virtual call is redirected, the call connected or clear indication packets sent to the calling DTE will contain the called address of the alternate DTE and the calling line address modified notification facility, indicating the reason why the called address is different from the one originally requested.

When the virtual call is redirected, some networks may indicate to the alternate DTE the reason for redirection and the address of the originally called DTE, using the call redirection notification facility in the incoming call packet.

The order of call set-up processing at the originally called DCE as well as the alternate DCE will be according to the sequence of call progress signals in Table 1/AX.96. For those networks that provide systematic call redirection with the prior re uest of the called DTE, the systematic call re8irection request will have the highest priority in the call set-up processing sequence at the originally called DCE.

7.2.10 Called line address modified notification

Called line address modified notification is a user facility, used by the DCE in the call connected or clear indication packets to inform the calling DTE as to why the called address, if present, in these packets is different from that specified in the call request packets.

The following reasons can be indicated with the use of called line address modified notification facility:

1) Call distribution within a Hunt Group.

2) Call redirection due to originally called DTE out of order.

3) Call redirection due to originally called DTE busy.

4) Call redirection due to prior request from the originally called DTE for systematic call redirection.

7.2.11 Call redirection notification

Call redirection notification is a user facility, used by the DCE in the incoming call packet to inform the alternate DTE as to why the call is redirected, and the address of the originally called DTE.

The following reasons can be indicated with the call redirection notification facility:

1) Call redirection due to originally called DTE out of order.

2) Call redirection due to originally called DTE busy.

3) Call redirection due to prior request from the originally called DTE for systematic call redirection.

7.2.12 Amateur networking facilities

The following describes optional amateur radio networking facilities. These facilities are interim recommendations, subject to corrections.

One of the two amateur routing facilities must be provided when connections are requested outside the local network.

7.2.12.1 Amateur network facility marker

The amateur related networking optional facilities must follow the CCITT optional facilities, and shall be separated from the CCITT facilities by a special two octet amateur marker.

7.2.12.2 Amateur explicit routing facility

The amateur explicit routing facility is an optional user facility that allows the calling DTE to specify the route of a connection.

When this facility is selected, the calling DTE must provide a list of addresses which will be used by the network to route the call to the called DTE.

7.2.12.3 Amateur implicit routing facility

The amateur implicit routing facility is an optional user facility that allows the network to route the call based on geographical information supplied by the calling DTE.

When this facility is selected, the calling DTE must provide the geographical location of the called DTE.

7.2.13 Additional optional facilities

In addition to the above facilities, others may be added as necessary. Subject for further study are certain X.75 utilities, such as call control.

7.3 Procedures for optional user facilities only available with Datagram service

These facilities are not implemented in AX.25.

7.4 Formats for optional user facilities

7.4.1 General

The facility field is present only when a DTE is using an optional user facility requiring some indication in the call request, incoming call, call accepted, call connected, clear request, or clear indication.

The facility field contains one of more facility elements. The first octet of each facility element contains a facility code to indicate the facility or facilities requested.

The facility codes are divided into four classes, by making use of bits 8 and 7 of the facility code field, in order to specify facility parameters consisting of 1, 2, 3, or a variable number of octets. The general class coding of the facility code field is shown in Table 15/AX.25 (Table 15/AX.25 is at the end of this paper).

For class D the octet following the facility code indicates the length, in octets, of the facility parameter field. The facility parameter field length is binary coded, with bit 1 being the low order bit.

The formats for the four classes are shown in Table 16/AX.25.

The facility code field is binary coded and, without extension, provides for a maximum of 64 facility codes for classes A, B, and C, and 63 facility codes for class D giving a total of 255 facility codes.

Facility code 11111111 is reserved for extension of the facility code. The octet following this octet indicates an extended facility code having the format A, B, C, or D as defined above. Repetition of facility code 11111111 is permitted and thus additional extensions may result.

The coding of the facility parameter field is dependent on the facility being requested.

A facility code may be assigned to identify a number of specific facilities, each having a bit in the parameter field indicating facility requested/facility not requested. In this situation, the parameter field is binary encoded with each bit position relating to a specific facility. A 0 indicates that the facility related to the particular bit is not requested and a 1 indicates that the facility related to the particular bit is requested. Parameter bits not assigned to a specific facility are set to 0. If none of the facilities represented by the facility code are requested for a virtual call, the facility code and its associated parameter field need not be present.

A facility marker, consisting of a single octet pair, is used to separate requests for AX.25 facilities, as defined in this section, from requests for non-AX.25 facilities that also may be offered by a network. The first octet is a facility code and is set to zero, and the second octet is the facility parameter field.

The coding of the parameter field will be either all zeros or all ones depending on whether the facility requests following the marker refer to facilities offered calling/source or called/destination network, respectively. For intranetwork virtual calls, the parameter field should be all zeros.

Requests for non-AX.25 facilities offered by the calling/source and called/destination networks may simultaneously present within the facility field and in such cases two facility markers will be required with parameter fields coded as described above.

Within the facility field, requests for AX.25 facilities will precede all requests for non-AX.25 facilities and facility markers need only be present when requests for non-AX.25 facilities are present.

```
         Class A                         Class B

bits 8 7 6 5 4 3 2 1            bits 8 7 6 5 4 3 2 1
     ------------------              ------------------
O 0 ! 0 0 X X X X X X !         O  0 ! 0 1 X X X X X X !
C   !-----------------!         C    !-----------------!
T   !    Facility     !         T  1 !    Facility     !
E 1 ! Parameter Field !         E    !---              !
T   ------------------          T  2 ! Parameter Field !
                                     ------------------

         Class C                         Class D

bits 8 7 6 5 4 3 2 1            bits 8 7 6 5 4 3 2 1
     ------------------              ------------------
O 0 ! 1 0 X X X X X X !         O  0 ! 1 1 X X X X X X !
C   !-----------------!         C    !-----------------!
T 1 !    Facility     !         T  1 !    Facility     !
E   !---           ---!         E    !---           ---!
T 2 !   Parameter     !         T  2 !   Parameter     !
    !---           ---!
  3 !     Field       !         V    !---  Field       !
    ------------------          A    !
                                R    ------------------
```

Table 16/AX.25
Facility Class Coding

7.4.2 Coding of facility field for particular facilities

7.4.2.4 Coding of RPOA selection facility

7.4.2.4.1 Facility code field

The coding of the facility code field for RPOA selection is:

bit:	8 7 6 5 4 3 2 1
code:	0 1 0 0 0 1 0 0

7.4.2.4.2 Facility parameter field

The parameter field contains the data network identification code for the requested RPOA transit network, and is in a form of four decimal digits.

Each digit is coded in a semi-octet (nibble) in binary coded decimal wit& bit 5 of the first octet being the low order bit of the first digit, bit 1 of the first octet being the low order bit of the second digit, bit 5 of the second octet being the low order bit of the third digit, and bit 1 of the second octet being the low order bit of the fourth digit.

7.4.2.5 Coding of flow control parameter facility negotiation

The coding of the facility code field and the format of the facility parameter field for packet sizes are the same in call request, incoming call, call accepted, and call connected packets.

7.4.2.5.1.1 Facility code field

The coding of the facility code for packet sizes is:

bit: 8 7 6 5 4 3 2 1
code: 0 1 0 0 0 0 1 0

7.4.2.5.1.2 Facility parameter field

The packet size for the direction of transmission from the called DTE is indicated in bits 4, 3, 2, and 1 of the first octet. The packet size for the direction of transmission from the calling DTE is indicated in bits 4, 3, 2, and 1 of the second octet. Bits 5, 6, 7, and 8 of each octet must be zero.

The four bits indicating each packet size are binary coded and express the logarithm base 2 of the number of octets of the maximum packet size.

Networks may offer values from 4 to 12, corresponding to packet sizes of 16, 32, 64, 128, 256, 512, 1024, 2048, or 4096, or a subset of these values. All networks should provide a packet size of 128.

7.4.2.5.2 Coding for window sizes

The coding of the facility code field and the format of the facility parameter field for window sizes are the same in call request, incoming call, call accepted, and call connected packets.

7.4.2.5.2.1 Facility code field

The coding of the facility code field for window sizes is:

bit: 8 7 6 5 4 3 2 1
code: 0 1 0 0 0 0 1 1

7.4.2.5.2.2 Facility parameter field

The window size for the direction of transmission from the called DTE is indicated in bits 7 to 1 of the first octet. The window size for the direction of transmission from the calling DTE is indicated in bits 7 to 1 of the second octet. Bit 8 of each octet must be zero.

The bits indicating each window size are binary coded and express the size of the window. A value of zero is not allowed.

Window sizes of 8 to 128 are not allowed in AX.25. All networks will provide a window size of 2.

7.4.2.7 Coding of fast select facility

The coding of the facility code and parameter fields for fast select is the same in call request and incoming call packets.

7.4.2.7.1 Facility code field

The coding of the facility code field for fast select is:

bit: 8 7 6 5 4 3 2 1
code: 0 0 0 0 0 0 0 1

7.4.2.7.2 Facility parameter field

The coding of the facility parameter field is:

Bit 8=0 and bit 7=0 or 1 for fast select not requested.

Bit 8=1 and bit 7=0 for fast select requested with no restriction on response.

Bit 8=1 and bit 7=1 for fast select requested with restriction on response.

Bits 6, 5, 4, 3, and 2 may be used for other facilities. If not they are set to zero. Use of bit 1 is described in section 7.4.2.3.

7.4.2.11 Coding of called line address modified notification

7.4.2.11.1 Facility code field

The coding of the facility code field for called line address modified notification is:

bit: 8 7 6 5 4 3 2 1
code: 0 0 0 0 1 0 0 0

7.4.2.11.2 Facility parameter field

The coding of the facility parameter field for called line address modified notification is:

bits: 8 7 6 5 4 3 2 1	
0 0 0 0 0 1 1 1	Call distribution within a Hunt Group.
0 0 0 0 0 0 0 1	Call redirection due to originally called DTE busy.
0 0 0 0 1 0 0 1	Call redirection due to originally called DTE out of order.
0 0 0 0 1 1 1 1	Call redirection due to prior request from originally called DTE for systematic call redirection.

7.4.2.12 Coding of call redirection notification

7.4.2.12.1 Facility code field

The coding of the facility code field for call redirection notification is:

bit: 8 7 6 5 4 3 2 1
code: 1 1 0 0 0 0 1 1

7.4.2.12.2 Facility parameter field

The octet following the facility code field indicates the length in octets of the facility parameter field and has the value n+2 where n is the number of octets necessary to hold the originally called DTE address.

The second octet indicates the reason for the call redirection and has one of the following values:

bits: 8 7 6 5 4 3 2 1	
0 0 0 0 0 0 0 1	Originally called DTE busy
0 0 0 0 1 0 0 1	Originally called DTE out of order
0 0 0 0 1 1 1 1	Systematic call redirection

The third octet indicates, in bits 4, 3, 2, and 1, the number of nibbles in the originally called DTE address. This address Length indicator is binary coded, with bit 1 being the low order bit. Bits 8, 7, 6, and 5 of this octet are set to zero.

The following octets (up to eight) contain the originally called DTE address, coded identically to the called DTE address field in the call request packet (see 6.2.1.3).

7.4.13 Coding of amateur network facilities

7.4.13.1 Coding of the amateur network facilities marker

The amateur network facilities marker field consists of two octets and is coded as folows:

bits: 8 7 6 5 4 3 2 1
octet1: 0 0 0 0 0 0 0 0
octet2: 1 1 1 1 1 1 1 0

7.4.13.2 Coding of amateur network explicit routing facility

7.4.13.2.1 Facility code field

The coding of the facility code field for the amateur network explicit routing is:

bits: 8 7 6 5 4 3 2 1
code: 1 1 0 0 0 0 0 0

7.4.13.2.2 Facility parameter field

The coding of the facility parameter field in the amateur network explicit routing facility is as follows:

The first octet is called the explicit length octet., and contains the total length of the facility, including the code field and the length octet. This length octet is binary coded, and cannot be zero.

The octets following the length octet contain the packet switch identifier(s) that indicate which switches the call should progress through.

The packet switch identifier consists of the six unshifted ASCII characters of the amateur callsign (upper case alpha and numeric characters only) of the packet switch, followed by an additional octet that contains a five-bit station subaddress. The station subaddress information is contained in bits 5, 4, 3, 2, and 1 of the seventh octet. Bits 8, 7, and 6 of the subaddress octet are reserved, and set to zero. If the callsign contains less than six ASCII characters, the callsign will be padded with trailing ASCII spaces between the last callsign character and the subaddress.

Up to 38 packet switch identifiers are allowed, the first identifier will identify the first packet switch in the chain. Additional switch identifiers following the first in order of call progress from the calling DTE to the called DTE.

Other methods of explicit routing are a subject for further study.

7.4.13.3 Coding of amateur network implicit routing-facility

7.4.13.3.1 Facility code field

The coding of the facility code field for the amateur network implicit routing facility is:

continued next column....

bits:	8 7 6 5 4 3 2 1
code:	1 0 0 0 0 0 0 1

7.4.13.3.2 Facility parameter field

The coding of the facility parameter field for the amateur network implicit routing facility consists of three octets.

The coding of this parameter is a subject for further study. The following information is one suggested method of coding this information, based on the geographical location of the called DTE.

The first octet contains the Longitude in degrees of the called station, from 0 to 180 based on Greenwich, England. This octet is binary encoded, with bit 8 being the most-significant bit, and bit 1 the least-significant bit.

The second octet contains the Latitude of the called station in degrees. Bit 8 indicates whether the called DTE is north or south of the equator (zero equals north, one equals south). Bits 7 to 1 contain the Latitude in degrees, from 0 to 89. This data is binary coded, with bit 7 being the MSB, and bit 1 is the LSB.

The third octet contains an east/west marker bit, and grid information within the Latitude/Longitude specified.

Bit 8 of the third octet contains the east/west marker for the Longitude information. If bit 8 is a zero, the Longitude information is west of Greenwich, while a one indicates the information is east of Greenwich.

The remaining seven bits contain information on where within the square (resulting from the Latitude/Longitude information above) the station is located. Actual coding of this field is a subject for further study.

Other methods of encoding the amateur network implicit routing facility are subject to further study. One method might be to use National Traffic System abbreviations. Comments or suggestions are welcome.

Facility indication	Valid facility request
W(indicated) =>2	W(indicated) =>W(requested) =>2
W(indicated) = 1	W(requested) = 1 or 2
P(indicated) =>128	P(indicated) =>P(requested)=>128
P(indicated) < 128	128 =>P(requested) =>P(indicated)

Table 13/AX.25
Valid facility requests in call accepted packets in response to facility indications in incoming call packets

Facility request	Valid facility indication
W(requested) =>2	W(requested) =>W(indicated) =>2
W(requested) = 1	W(indicated) = 1 or 2
P(requested) =>128	P(requested) =>P(indicated)=>128
P(requested) < 128	128 =>P(indicated) =>P(requested)

Table 14/AX.25
Valid facility indications in call connected packets in response to facility requests in call requests packets

Class	8 7 6 5 4 3 2 1	parameter
Class A	0 0 X X X X X X	for single octet parameter
Class B	0 1 X X X X X X	for double octet parameter
Class C	1 0 X X X X X X	for triple octet parameter
Class D	1 1 X X X X X X	for variable lenght param.

Where X can be either a zero or one.

Table 15/AX.25
Facility Class Coding

Terry Fox, WB4JFI
President, AMRAD
1819 Anderson Rd.
Falls Church, VA 22043

Description

This is the fourth of four papers that make up a recommendation for the AX.25 Network Sublayer protocol.

This paper contains the annexes for the previous three papers. These annexes are based on the CCITT X.25 document, modified as necessary to operate in the amateur enviroment.

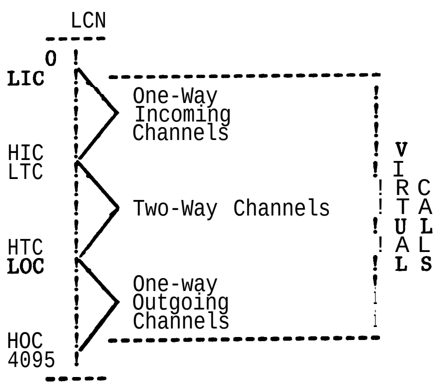
This paper is a draft, and subject to change. Anyone wishing to offer comments or suggestions should write the author at the above address, or write to the AMRAD Newsletter for publication.

ANNEX A

Range of logical channels used for virtual calls

In the case of a single logical channel DTE, logical channel number 1 will be used.

For each multiple logical channel DTE/DCE interface, a range of logical channels will be agreed upon with the Network Adminidtration according to Figure A-1/AX.25.



Where:

LCN = Logical channel number
LIC = Lowest Incoming call number
HIC = Highest Incoming call number
LTC = Lowest Two-way call number
HTC = Highest Two-way call number
LOC = Lowest Outgoing call number
HOC = Highest Outgoing call number

Figure A-1/AX.25

LIC to HIC: range of logical channels assigned to one-way incoming channels for virtual calls.

The present recommendation is to assign 1 as the LIC, and 3 as the HIC.

LTC to HTC: range of logical channels which are assigned to two-way logical channels for virtual calls.

The present recommendation is to assign 4 as the LTC, and 4079 as the HTC.

LOC to HOC: range of logical channels assigned for use as one-way outgoing channels for virtual calls.

The present recommendation is to assign 4080 as the LOC, and 4095 as the HOC.

Note 1: The reference to the number of logical channels is made according to a set of contiguous numbers from 0 (lowest) to 4095 (highest) using 12 bits made up of the logical channel group number (LCGN) (see 6.1.2) and the 8 bits of the logical channel number (see 6.1.3). The numbering is binary coded using bit positions 4 to 1 of octets 1 followed by bit positions 8 through 1 of octet followed by bit positions 8 through 1 of octet 2 with bit 1 of octet being the low order bit.

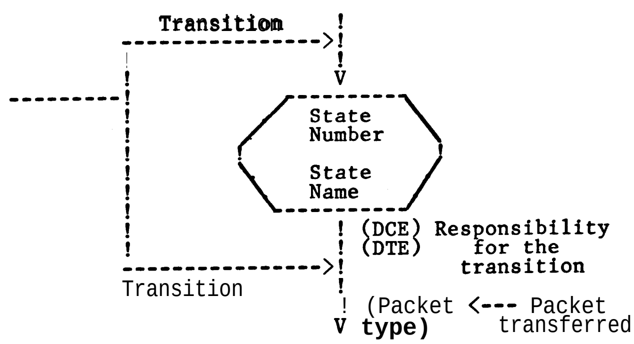
Note 2: All logical channel boundaries are agreed to for a period of time.

Note 3: DCE search algorithm for a logical channel for a new incoming call will be to use the lowest logical channel in the ready state in the range of LIC to HIC or LTC to HTC, depending on whether the call is one-way incoming or two-way, respectively.

Note 4: In order to minimize the risk of call collision, the DTE search algorithm is suggested to start with the highest numbered logical channel in the ready state.

ANNEX B

B.1 Symbol definition of the state diagram



Note 1. Each state is represented by an ellipse wherein the state name and number are indicated.

Note 2. Each state transition is represented by an arrow. The responsibility for the transition (DTE or DCE) and the packet that has been transferred as indicated beside the arrow.

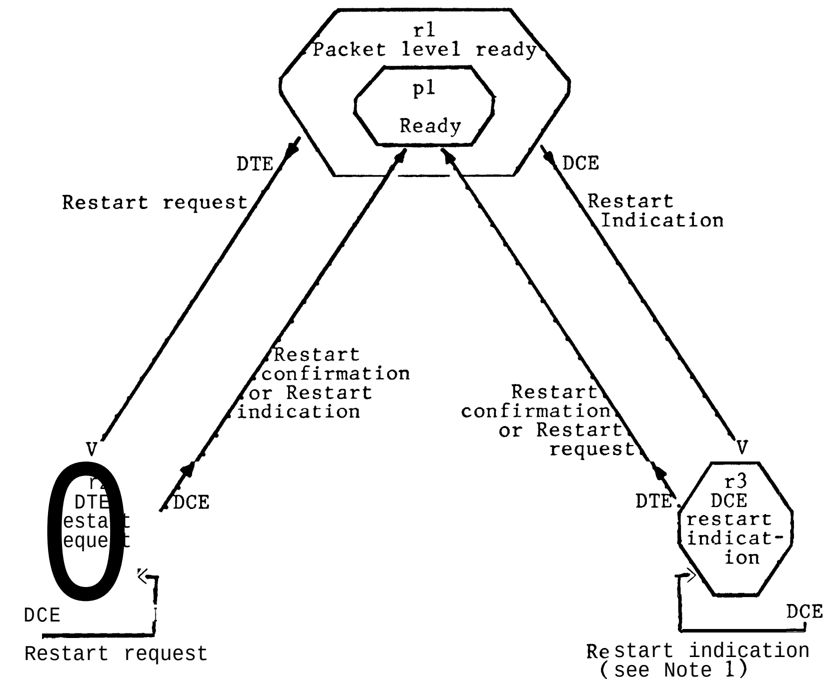
B.2 Order definition of the state diagram

For the sake of clarity, the normal procedure at the interface is described in a number of small state diagrams. In order to describe the normal procedure fully, it is necessary to allocate a

priority co the different figures and to relate a higher order diagram with a lower one. This has been done by one of the following means:

The figures have been arranged in order of priority with Figure B-l/AX.25 (restart) having the highest priority and subsequent figures having lower priority. Priority means that when a packet belonging to a higher order diagram is transferred, that diagram is applicable and the lower one is not.

The relation with a state in a lower order diagram is given by including that state inside an ellipse in the higher order diagram.



Note 1. This transition may take place sfter time-out T10.

Figure B-l/AX.25

Diagram of states for the transfer of restart packets

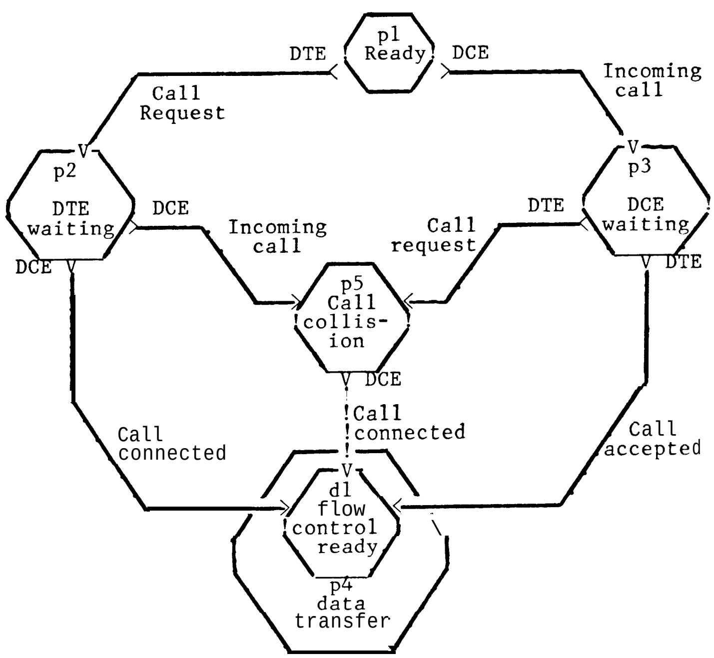
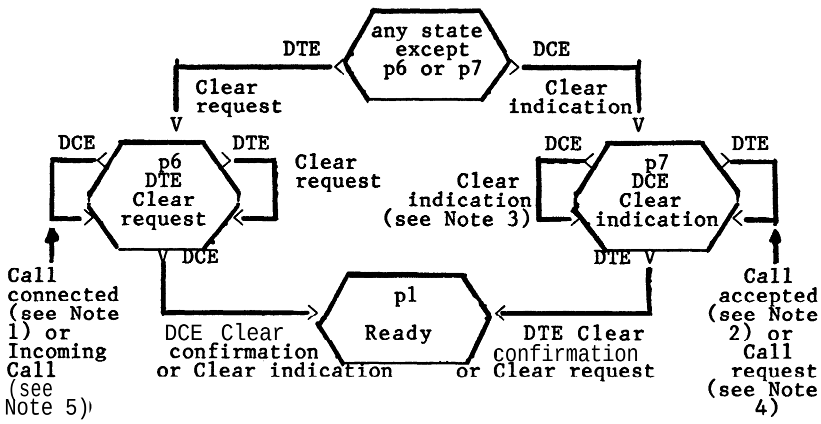


Figure B-2AfAX.25
Call set-up phase



Note 1. This transition is possible only if the previous state was DTE waiting (p2).

Note 2. This transition is only possible if the previous state was DCE waiting (p3).

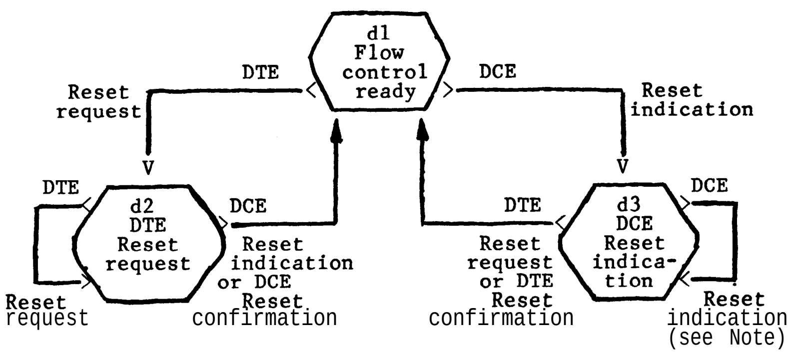
Note 3. This transition may take place after time-out T13.

Note 4. This transition is possible only if the previous state was Ready (p1) or DCE waiting (p3).

Note 5. This transition is possible only if the previous state was Ready (p1) or DTE waiting (p2).

Figure B-2B/AX.25
call clearing phase

Diagrams of states for the transfer of call set-up and call clearing packets within the packet level ready (p1) state.



Note: this transition may take place after time-out T12

Figure B-3/AX.25

Diagram of states for the transfer of reset packets within the data transfer (p4) state

ANNEX C

Actions taken by the DCE on receipt of packets in a given state of the packet level DTE/DCE as perceived by the DCE

INTRODUCTION

This Annex specifies the actions taken by the DCE on receipt of packets in a given state of the packet level DTE/DCE interface as perceived by the DTE.

It is presented as a succession of chained tables.

The following rules are valid for all these tables:

1) There may be more than one error associated with a packet. The network will stop normal processing of a packet when an error is encountered. Thus, only one diagnostic code is associated with an error indication by the DCE. The order of packet decoding and checking on networks is NOT standardized.

2) The detection of the non-octet alignment shall be made at the frame level.

3) In each table, tha actions taken by the DCE are indicated in the following way:

A) Discard: The DCE discards the received packet and takes no subsequent action as a direct result of receiving that packet, the DCE remains in the same state.

B) DIAG # x: The DCE discards the received packet and, for networks which implement the diagnostic packet, transmits to the DTE a diagnostic packet containing the diagnostic # x.

C) NORMAL or ERROR: The corresponding action is specified behind each table.

4) Annex E gives a list of the diagnostic codes which may be used.

Table C-1/AX.25
Special Cases

Packet from the DTE	Any State
Any packet with packet length shorter than two octets	DIAG #38
Any packet with an incorrect general format identifier (GFI)	DIAG #40
Any packet with an unassigned logical channel	DIAG #36
Any packet with correct GFI and assigned logical channle or with bits 1 to 4 of octet 1 and bits 1 to 8 of octet 2 equal to zero	See Table C-2/AX.25

State of the interface as perceived by the DTE / Packet from the DTE with assigned logical channel	Packet Level Ready r1	DTE Restart Request r2	DCE Restart Indication r3
Restart Request	Normal (r2)	Discard	Normal (r1)
DTE Restart Confirmation	Error #17 (r3)	Error #18 (r3)	Normal (r1)
Data, interrupt, call set-up and clearing, flow control or reset	see Table C-3/AX.25	Error #18 (r3)	Discard
Restart request or DTE restart confirmation with bits 1 to 4 of octet 1 or bits 1 to 8 octet 2 unequal to zero	see Table C-3/AX.25	Error #41 (r3)	Discard
Packets having a packet type identifier which is shorter than 1 octet or is not supported by the DTE	see Table C-3/AX.25	Error #38 Error #33 (r3)	Discard
Packet other than restart request and DTE restart confirmation with bits 1 to 4 of octet 1 and bits 1 to 8 of octet 2 equal to 0	DIAG # 36	DIAG # 36	DIAG # 36

TABLE C-2/AX.25
Action taken by the DCE on receipt of packets in a given state of the packet level DTE/DCE interface as perceived by the DCE: restart procedure for assigned logical channels.

Where:

Error (r3): The DCE discards the received packet, indicates a restarting by sending to the DTE a restart indication packet with the cause "Local procedure error" and the diagnostic # x, and enters state r3. If connected through a virtual call, the distant DTE is also informed of the restarting by a clear indication packet, with the cause "Remote procedure error" (same diagnostic).

If a restart indication is issued as a result of an error condition in state r2, the DCE should eventually (after a time which does not exceed 120 seconds) consider the DTE/DCE interface to be in the packet level ready state (r1).

NORMAL (r1): Provided none of the following error conditions has occured, the action taken by the DCE follows the procedure as defined by in section 3, and the DTE/DCE interface enters state rl:

a) If a restart request packet or DTE restart confirmation packet received in state r3 exceeds the maximum permitted length or is too short, the DCE will invoke the ERROR #39 or #38 procedure, respectively.

b) If a restart request packet received in state rl exceeds the maximum permitted length, is too short, or the restarting cause field is not "DTE originated", the DCE shall invoke the DIAG #32, 838, or #81.

State of the interface as perceived by the DTE / Packet from the DTE with logical assigned to the virtual call	Ready p1	DTE Waiting p2	DCE Waiting p3	Data Transfer p4	Call Collision p5	DTE Clear request p6	DTE Clear indication p7
Call Request	Normal (p2)	Error #27 (p7)	Normal (p5)	Error #23 (p7)	Error #24 (p7)	Error #25 (p7)	Discard
Call Accepted	Error #20 (p7)	Error #21 (p7)	Normal (p4)	Error #23 (p7)	Error #24 (p7)	Error #25 (p7)	Discard
Clear Request	Normal (p6)	Normal (p6)	Normal (p6)	Normal (p6)	Normal (p6)	Discard	Normal (p6)
DTE Clear Confirmation	Error #20 (p7)	Error #21 (p7)	Error #22 (p7)	Error #23 (p7)	Error #24 (p7)	Error #25 (p7)	Normal (p1)
Data, Interrupt, Reset or Flow Control	Error #20 (p7)	Error #21 (p7)	Error #22 (p7)	see Table C-4/AX.25	Error #24 (p7)	Error #25 (p7)	Discard
Restart request, or DTE restart confirmation with bits 4 to 1 of octet 1 or bits 1 to 8 of octet 2 unequal to zero	Error #41 (p7)	Error #41 (p7)	Error #41 (p7)	see Table C-4/AX.25	Error #41 (p7)	Error #41 (p7)	Discard
Packets having a packet type identifier which is shorter than one octet or is not supported by the DCE	Error #38 #33 (p7)	Error #38 #33 (p7)	Error #38 #33 (p7)	see Table C-4/AX.25	Error #38 #33 (p7)	Error #38 #33 (p7)	Discard

TABLE C-3/AX.25

Action taken by the DCE on repeipt of packets in a given state of the packet level DTE/DCE interface as perceived by the DCE: call set-up and clearing on logical channel assigned to the virtual call.

Where:

Error (p7): The DCE discards the received packet, indicates a clearing by transmitting to the DTE a clear indication packet, with the "Local procedure error" and the diagnostic # x, and enters state p7. If connected through a virtual call, the distane DTE is also informed of the clearing by a clear indication packet, with the cause "Remote procedure error" (same diagnostic).

It is required that in the absence of an appropriate DTE response to a clear indication packet issued as a result of an error condition in state p6, the DCE should eventually consider (after a time which does not exceed 120 seconds) the DTE/DCE interface to be in the ready state (p1).

Normal (p1): Provided none of the following error conditions has occured, the action taken by the DCE follows the procedures as defined in section 4 and the DTE/DCE interface enters state pl. In all the cases specified hereunder, the DCE will transmit to the DCE a clear indication with the appropriate cause and diagnostic, and enter state p7; if connected through a virtual call, the distant DTE is also informed of the clearing by a clear indication packet with the cause "Remote procedure error" (same diagnostic).

a) Call request packet

Error Condition	Cause	Specific diagnostics (see Note 3 of ANNEX E)
1. Not applicable		
2. not applicable		
3. Address contains a non-BCD digit	Local procedure error	# 67,68
4. Prefix digit not supported	Local procedure error	# 67,68
5. National address smaller than national address format permits	Local procedure error	# 67,68
6. National address larger than national address format permits	Local procedure error	# 67,68

Error Condition	Cause	Specific diagnostics (see Note 3 of ANNEX E)
7. DNIC less than four digits	Local procedure error	# 67,68
8. bits 7 or 8 of octet which indicates the facility field length not set to zero	Local procedure error	# 69
9. no combination of facilities could equal facility length	Local procedure error	# 69
10. Facility length larger than remainder of packets	Local procedure error	# 38
11. Facility values conflicts (ex. a particular combination not supported).	Invalid facility request	# 66
12. Facility code not allowed.	Invalid facility request	# 65
13. Facility value not allowed	Invalid facility request	# 66
14. Packet too short	Local procedure error	# 38
15. Address length larger than remainder of packet	Local procedure error	# 38
16. Call user data larger than 16, or 128 in case of fast select facility	Local procedure error	# 39
17. Class coding of the facility corresponding to a length of parameter larger than remainder of packet	Local procedure error	# 69
18. Facility code repeated	Local procedure error	# 73

If a virtual call cannot be established by the network, the DCE should use a call progress signal and diagnostic code among those listed hereunder:

Error Condition	Cause	Specific diagnostics (see Note 3 of ANNEX E)
19. Unknown station	Not obtainable	# 67
20. Not applicable		
21. Not applicable		
22. Reverse charging rejected	Reverse charging acceptance not subscribed	# 0
23. Fast select rejected	Fast select acceptance not sub scribed	# 0
24. Called DTE out of order	Out of order	# 0 # greater than 128
25. No logical channel available	Number busy	# 71
26. Call collision	Number busy	# 71,72
27. RPOA out of order	RPOA out of order	# 0
28. Temporary network congestion of fault condition within the network	Network congestion	# 0
29. The remote DTE/DCE interface or the transit network does not support a function or a facility requested	Incompatible destination	# 0
30. Procedure error at the remote DTE/DCE interface	Remote procedure error	see b and c and ANNEX D

b) Call accepted packet

Error condition	Cause	Specific Diagnostics (see Note 3 of Annex E)
1. Not applicable		
2. Address contains a non-BCD digit	Local procedure error	# 67,68
3. Prefix digit not supported	Local procedure error	# 67,68
4. National address smaller than national address format permits	Local procedure error	# 67,68
5. National address larger than national address format permits	Local procedure error	# 67,68
6. DNIC less than four digits	Local procedure error	# 67,68
7, bits 7 or 8 of octet which indicates the facility field length not set to zero	Local procedure error	# 69
8. no combination of facilities could equal facility length	Local procedure error	# 69
9. Facility length larger than remainder of packets	Local procedure error	# 38
10. Facility values conflicts (ex. a particular combination not supported).	Invalid facility request	# 66
11, Facility code not allowed.	Invalid facility request	# 65
12, Facility value not allowed or invalid	Invalid facility request	# 66
13. Address length larger than remainder of packet	Local procedure error	# 38
14, Call user data larger than 128 in case of fast select facility	Local procedure error	# 39
15. Class coding of the facility corresponding to a length of parameter larger than remainder of packet	Local procedure error	# 69

16. Facility code repeated	Local procedure error	# 73
17. The incoming **call packet** indicated fast select with **restriction on response**	**Local procedure error**	**# 42**

c) Clear request packet

Error condition	Cause	Specific Diagnostics (see Note 3 of Annex E)
1. Not applicable		
2. Packet **too short**	**Local** procedure **error**	**# 38**
3. Packet length larger than 5 octets (if fast select facility not requested)	**Local procedure error**	# 39
4. Non zero address length field (if fast select facility requested)	**Local procedure error**	**# 74**
5. Non zero facility field (if fast select facility requested)	**Local procedure error**	# 75
14. Call user data larger than 128 in case of fast select facility (if fast **select facility requested)**	**Local procedure error**	# 39
15. Clearing cause field is not **"DTE** originated" in the **clear request packet**	**Local procedure** error	**# 81**

d) DTE **clear confirmation**

Error Condition	**Cause**	Specific diagnostics (see Note 3 of ANNEX E)
1. Not applicable		
2. Packet length **larger than 3 octets**	**Local procedure** error	**# 39**

State of the interface as perceived by the DCE / Packet from the DTE with assigned logical channel	Data Transfer p4		
	Flow Control Ready d1	DTE Reset Request d2	DCE Reset Indication d3
Reset Request	Normal (d2)	Discard	Normal (d1)
DTE Reset Confirmation	Error #27 (d3)	Error #28 (d3)	Normal (d1)
Data, interrupt, or flow control	Normal (d1)	Error #28 (d3)	Discard
Restart request or DTE restart confirmation with bits 1 to 4 of octet 1 or bits 1 to 8 octet 2 unequal to zero	Error #41 (d3)	Error #41 (d3)	Discard
Packets having a packet type identifier which is shorter than 1 octet or is not supported by the DCE	Error #38 Error #33 (d3)	Error #38 Error #33 (d3)	Discard

TABLE C-4/AX.25
Action taken by the DCB on receipt of packets in a given state of the packet level DTE/DCE interface as perceived by the DCE: data transfer (flow control) on assigned logical channels

Error **(d3):** The DCE discards the received packet, indicates a reset bytransmittingto the DTE a reset indication packet with the cause "Local **procedure** error* and the diagnostic **# x,** and enter state d3. For virtual calls, the distant DTE is also informed of the reset by a reset indication packet, with the cause "Remote procedure error" (same diagnostic).

If a reset indication is issued as a result of an error condition in state **d2,** the DCE should eventually consider (after a time not to exceed 120 seconds) the DTE/DCE interface to be in the flow control ready state (dl).

Provided none of the following error conditions has **occured,** the action taken by the DCE follows the procedure as described in sections 4 and 5:

a) If the packet exceeds the maximum permitted length, or is too short, the DCE will invoke the Error # 39 **or # 38** procedure, respectively.

b) If the P(S) or P(R) received is not valid, the DCE will invoke the Error # 1 or # 2 procedure, respectively.

c) The DCE will consider the receipt of a DTE interrupt confirmation packet which does not correspond to a yet unconfirmed DCE interrupt packet as an error and will invoke the Error # 43 procedure. The DCE will either discard or consider as an error a DTE interrupt packet received before a previous DTE interrupt packet has been confirmed (Error # 44 procedure).

d) Not applicable

e) If the resetting cause field is not **"DTE** originated" in a reset **request packet,** the error procedure is **invoked**. A reset indication packet will be* transmitted with the cause "Local procedure error" and the diagnostic # 81.

ANNEX D

Packet level DCE time-outs and DTE time-outs

D.1 DCE time-outs

Under certain circumstances, this recommendation requires the DTE to respond to a packet issued from the DCE within a stated maximum time.

Table D-1/AX.25 covers these circumstances and the actions that the DCE will initiate upon the expiration of that time.

D.2 DTE time-outs

Under certain cirmstances, this Recommendation requires the DCE to respond to a packet from the DTE within a stated maximum time. Table D-Z/AX.25 gives these maximum times. The actual DCE response times should be well within the specified time limits. The rare situation where a time-limit is exceeded should only occur when there is a fault condition.

To facilitate recovery from such fault conditions, the DTE may incorporate timers. The time-limits given in Table D-2/AX.25 are the lower limits of the times a DTE should allow for proper operation. Suggestions on possible DTE actions upon expiration of the time-limits are given in Table D-2/AX.25.

Time-out No.	Time-out value	State of the logical channel	Started When	Normally Terminated when	Actions to be taken when the time-out expires: Local side	Actions to be taken when the time-out expires: Remote side
T10	60 secs.	r3	DCE issues a restart indication packet	DCE leaves the r3 state (ex: the restart confirmation or restart request is received)	DCE remains in r3 and may issue a diagnostic packet	DCE enters the d3 state signalling a reset indication (remote procedure error)
T11	180 secs.	p3	DCE issues an incoming call packet	DCE leaves the p3 state (ex: the call accepted, clear request, or call request is received	DCE enters the p7 state signalling a clear indication (local procedure error)	DCE enters the p7 state signalling a clear indication (remote procedure error)
T12	60 secs.	d3	DCE issues a reset indication packet	DCE leaves the d3 state (ex: the reset confirmation or reset request is received)	For virtual calls, the DCE enters the p7 state signalling a clear indication (local procedure error)	For virtual calls, DCE enters the p7 state signalling a clear indication (remote procedure
T13	60 secs.	p7	DCE issues a clear indication packet	DCE leaves the p7 state (ex: the clear confirmation or clear request is received)	DCE remains in p7 and may issue a diagnostic packet	

Table D-1/AX.25. DCE Time-limits

Time-out number	Time-limit value	State of the logical Channel	Started When	Normally terminated When	Preferred action to be taken when time-limit expires
T20	180 secs.	r2	DTE issues a restart request packet	DTE leaves the r2 state (ex: the restart confirmation or restart indication is received)	To retransmit the restart request packet (see Note 1)
T21	200 secs.	p2	DTE issues a call request packet	DTE leaves the p2 state (ex: the call connected, clear indication, or incoming call is received	To transmit a clear request packet
T22	180 secs.	d2	DTE issues a reset request packet	DTE leaves the d2 state (ex: the reset confirmation or reset indication is rcvd)	For virtual calls, to retransmit the reset request or to transmit a clear request packet
T23	180 secs.	P6	DTE issues a clear request packet	DTE leaves the p6 state (ex: the clear confirmation or clear indication is rcvd)	To retransmit the clear request packet (see Note 2)

Note 1: After unsuccessful retries, recovery decisions should be taken at higher levels.

Note 2: After unsuccessful retries, the logical channel should be considered out-of-order. The restart procedure should only be invoked for recovery if reinitialization of all logical channels is acceptable.

Table D-2/AX.25. DTE Time-limits

ANNEX E

Coding of AX.25 network generated diagnostic fields in clear, reset, and restart indication and diagnostic packets (see Notes 1, 2, and 3)

Diagnostics	Bits 8 7 6 5 4 3 2 1	Decimal	Hex
No additional information	0 0 0 0 0 0 0 0	0	00
Invalid P(S)	0 0 0 0 0 0 0 1	1	01
Invalid P(R)	0 0 0 0 0 0 1 0	2	02
Packet type invalid	0 0 0 1 0 0 0 0	16	10
For state r1	0 0 0 1 0 0 0 1	17	11
For state r2	0 0 0 1 0 0 1 0	18	12
For state r3	0 0 0 1 0 0 1 1	19	13
For state p1	0 0 0 1 0 1 0 0	20	14
For state p2	0 0 0 1 0 1 0 1	21	15
For state p3	0 0 0 1 0 1 1 0	22	16
For state p4	0 0 0 1 0 1 1 1	23	17
For state p5	0 0 0 1 1 0 0 0	24	18
For state p6	0 0 0 1 1 0 0 1	25	19
For state p7	0 0 0 1 1 0 1 0	26	1A
For state d1	0 0 0 1 1 0 1 1	27	1B
For state d2	0 0 0 1 1 1 0 0	28	1C
For state d3	0 0 0 1 1 1 0 1	29	1D
(not implemented)	0 0 0 1 1 1 1 0	30	1E
	0 0 0 1 1 1 1 1	31	1F
Packet not allowed	0 0 1 0 0 0 0 0	32	20
Unidentifiable packet	0 0 1 0 0 0 0 1	33	21
(not implemented)	0 0 1 0 0 0 1 0	34	22
(not implemented)	0 0 1 0 0 0 1 1	35	23
Packet on unassigned logical channel	0 0 1 0 0 1 0 0	36	24
(not implemented)	0 0 1 0 0 1 0 1	37	25
Packet too short	0 0 1 0 0 1 1 0	38	26
Packet too long	0 0 1 0 0 1 1 1	39	27
Invalid General Format Ident	0 0 1 0 1 0 0 0	40	28
Restart with non-zero in bits in LCGN or LCN	0 0 1 0 1 0 0 1	41	29
Packet type not compatible with facility	0 0 1 0 1 0 1 0	42	2A
Unauthorized interrupt confirmation	0 0 1 0 1 0 1 1	43	2B
Unauthorized interrupt	0 0 1 0 1 1 0 0	44	2C
(not implemented)	0 0 1 0 1 1 0 1	45	2D
(not implemented)	0 0 1 0 1 1 1 0	46	2E
	0 0 1 0 1 1 1 1	47	2F
Time expired	0 0 1 1 0 0 0 0	48	30
For incoming call	0 0 1 1 0 0 0 1	49	31
For clear indication	0 0 1 1 0 0 1 0	50	32
For reset indication	0 0 1 1 0 0 1 1	51	33
For restart indication	0 0 1 1 0 1 0 0	52	34
	0 0 1 1 1 1 1 1	63	3F
Call set-up or clearing problems-	0 1 0 0 0 0 0 0	64	40
Facility code not allowed	0 1 0 0 0 0 0 1	65	41
Facility parameter not allowed	0 1 0 0 0 0 1 0	66	42
Invalid called address	0 1 0 0 0 0 1 1	67	43
Invalid calling address	0 1 0 0 0 1 0 0	68	44
Invalid facility length	0 1 0 0 0 1 0 1	69	45
(not implemented)	0 1 0 0 0 1 1 0	70	46
No logical channel available	0 1 0 0 0 1 1 1	71	47
Call collision	0 1 0 0 1 0 0 0	72	48
Duplicate facility requested	0 1 0 0 1 0 0 1	73	49
Non-zero address length	0 1 0 0 1 0 1 0	74	4A
Non-zero facility length	0 1 0 0 1 0 1 1	75	4B
non-assigned up to:	0 1 0 0 1 1 1 1	79	4F
Miscellaneous	0 1 0 1 0 0 0 0	80	50
Improper cause code from DTE	0 1 0 1 0 0 0 1	81	51
(not implemented)	0 1 0 1 0 0 1 0	82	52
Inconsistant Q bit setting	0 1 0 1 0 0 1 1	83	53
Maintenance action	0 1 0 1 0 1 0 0	84	54
non-assigned up to:	0 1 0 1 1 1 1 1	95	5F
Not assigned from :	0 1 1 0 0 0 0 0	96	60
to:	0 1 1 1 1 1 1 1	127	7F
Reserved for network specific diagnostic information	1 0 0 0 0 0 0 0	128	80
	1 1 1 1 1 1 1 1	255	FF

Note 1: Not all diagnostic codes need apply to a specific network, but those used are coded as shown in the table.-

Note 2: A given diagnostic need not apply to all packet types (ex: reset indication, clear indication, restart indication, and diagnostic packets)

Note 3: The first diagnostic in each grouping is a generic diagnostic and is used when'more specific diagnostics are not defined within the grouping. The decimal 0 diagnostic code can be used in situations where no other diagnostic applies.

ANNEX F

Address coding techniques for AX.25

Background

The following information will be called Appendix 1 of AX.25 in the future, in order to prevent conflicts with CCITT additions.

Address field description in AX.25

The following restrictions apply to the address field of X.25 packets (Section 5.2.3.2.2).

When present, octet 7 and the following octets consist of the called DTE address when present, then the calling DTE address when present.

Each digit of an address is coded in a semi-octet in binary-coded-decimal (BCD) with bit 5 or bit 1 being the low order bit of the digit.

Starting from the high order digit, the address is coded in octet 7 and consecutive octets with two digits per octet. In each octet, the higher order digit is coded in bits 8, 7, 6, and 5.

The address field shall be rounded up to an integral number of octets by inserting zeros in bits 4, 3, 2, and 1 of the last octet of the field when necessary.

Data Network Identification Code

CCITT recommendation X.121 specifies a method of creating an international numbering plan for public data networks. Part of this recommendation specifies the assignment of a four digit number to identify public data networks. This number is called the Data Network Identification Code, or DNIC.

The first three digits of the DNIC is used as a country code, and is called the Data Country Code (DCC). The fourth digit is used to identify the public network within the country, and is called the network digit.

The CCITT has the responsibility for assigning the DCC, a list of assigned DCC numbers is listed in X.121. The first DCC for the United States is 310.

The responsibility for assigning network digits is left to the responsible body within the country in question. The Federal Communications Commission is the responsible authority in the United States. Unfortunately, the FCC is not assigning network digits at the moment, so the amateurs are unable to have assigned a DNIC code to us for now. We will attempt to have assigned to the amateur network a DNIC number when possible.

For now, room should be left for the DNIC number, primarily to allow internetworking with existing public data networks.

Original DTE address techniques for AX.25

When the AX.25 draft committee originally met a method of coding the amateur station callsign into the DTE calling and DTE called fields of call request, incoming call, call accepted, and call connected packets. This involved coding the Data Network Identification Code (DNIC), a station subaddress, and the amateur call of each DTE into seven octets as follows.

The DNIC is a four digit number, and as such, would fit into the first two octets. The first semi-octet of the first octet would carry the first digit, with the three succeeding digits in the next three semi-octets.

The third octet would contain a five bit field used as a substation address. This field would be binary coded in bit positions 1 thru 5, with bit 1 being the LSB. Bits 6, 7, and 8 are reserved at this time, and set to zero.

The fourth through seventh octets contain the amateur station callsign. Since there is not enough room to contain the callsign directly, is was recommended that the callsign be coded so that up to six callsign characters could be fit into the four octets using Radix 50 coding. Radix 50 coding allows three upper-case alpha or numeric characters to fit into a six octal digit field. The fourth octet would contain the first portion of the radix-50 encoded characters, with succeeding octets carrying the rest of the information in order. If a callsign contained less than six ASCII characters, trailing ASCII space characters would be added as necessary.

It should be noted that the above method of coding could create illegal (not BCD)addressing information, which could cause problems at an interface to a public data network.

New method of addressing in AX.25 packets

Information has just reached me (as I am printing this paper) that at a recent meeting, the CCITT has added new methods of coding address information into X.25 packets. Some of the additions follow immediately, then comments by the author on how to use these new methods.

AX.25 Additions

Facility Markers

In the third paper of this series, under section 7.4.1, delete the last four paragraphs, and add the following:

In addition to the facility/registration codes defined in section 7, other codes may be used for :

-non-AX.25 facilities possibly provided by some network(s) (call set-up packets)

-CCITT-specified DTE facilities as described in Annex G of this recommendation (call set-up, clear request and clear indication packets).

Facility/registration markers, consisting of a single octet pair, are used to separate requests for AX.25 facilities as defined in sections 6 and 7 from other categories as defined above, and, when several categories of facilities are simultaneously present, to separate these categories from each other.

The first octet of the marker is a facility/registration code and is set to zero. The second octet is a facility/registration parameter field.

The facility registration parameter field of a marker is set to zero when the marker precedes requests for:

-non-AX.25 facilities provided by the network in case of intranetwork calls (call set-up packets).

-non-AX.25 facilities provided by the network to which the calling DTE is connected, in case of intranetwork calls.

The facility parameter field of a marker is set to all ones when the marker precedes requests for non-AX.25 facilities provided by the network to which the called DTE is connected, in case of intranetwork calls (call set-up packets).

The facility parameter field of a marker is set to 00001111 when the marker precedes requests for CCITT-specified DTE facilities.

All networks will support the facility markers with a facility parameter field set to all ones or 00001111.

DTEs should not use a facility marker with a facility parameter field set to all ones in case of intranetwork calls. however, if a DTE uses such a marker in an intranetwork call, the DCE is not obliged to clear the call, and the marker, with the corresponding facility requests, may be transmitted to the remote DTE.

Facility/registration codes for AX.25 facilities and for the other categories of facilities may be simultaneously present. However, requests for AX.25 facilities must

precede the other requests, and requests for CCITT-specified DTE facilities must follow the other requests.

The coding of CCITT-specified DTE facilities should comply with the description in Annex G. However, it is not required for the DCE to verify that compliance. If the network verifies that compliance and finds an error, it may clear the call. The CCITT-specified DTE facilities are passed unchanged between the two packet-mode **DTEs.**

(end of addition to7.4.1)

ANNEX G

CCITT-specified DTE facilities to support the OSI network service

G.l Introduction

The facilities described in this Annex are intended to support end-to-end signalling required by the **OSI** network service. They follow the CCITT-specified DTE facility marker defined in section 7.4. These facilities are passed are passed unchanged between the two packet mode **DTEs** involved.

Procedures for use of these **facilities by DTEs require definition by international user bodies.** Subsequent **provision** of X.25 **facilities** to be **acted** on by **public** data networks is for further study. Coding for these facilities is defined here in order to facilitate a consistent facility coding scheme in such future evolution.

G.2 **Coding of** the CCTTT-specified facilities

G.2.1 Calling address extension facility

The calling address extension facility is used in call request and incoming call packets to convey additional calling DTE address information.

G.2.1.1 Coding of the facility code field

The coding of the facility code field for the calling address extension facility is:

```
bits:   8 7 6 5 4 3 2 1
code:   1 1 0 0 1 0 0 0
```

Coding of the facility parameter field

The octet following the facility code field indicates the length in octets of the facility parameter field and has a value of n + 1, where n may be a maximum of 16 octets in order to hold the calling address extension.

The first octet of the facility parameter field indicates, in bits 6, 5, 4, 3, 2, and 1, the number of semi-octets (up to 32) in the calling address extension. This address length indicator is binary coded, and bit 1 is the low order bit. Bits 8 and 7 of this octet are set to zero.

The following octets (up to 16) contain the calling address extension.

Each digit of an address is coded in a semi-octet in binary coded decimal, where bit 5 or 1 is the low order bit of the digit.

Starting from the high-order digit, the address is coded in octet 2 and consecutive octets of the facility parameter field with two digits paer octet. In each octet, the higher order digit is coded in bits 8, 7, 6, and 5.

When necessary, the facility parameter field shall be rounded up to an integral number of octets by **inserting** zeros in bits 4, 3, 2, and 1 of the last octet of the field.

G.2.2 Called address extension facility

The called address extension **facility is** used in call request, incoming **call, call** accepted, call connected, clear indication, and clear request **packets** to convey additional called DTE address information.

G.2.2.1 **Coding** of the facility code field

The coding of the **facility** code field for the called address extension **facility** is:

```
bits:   8 7 6 5 4 3 2 1
code:   1 1 0 0 1 0 0 1
```

Coding of the facility parameter field

The octet following **the** facility code field indicates the length in octets of the facility parameter field and has a value of n + 1, where n may be a maximum of 16 octets in order to hold the called address extension.

The first octet of the facility parameter field indicates, in bits 6, 5, 4, 3, 2, and 1, the number of semi-octets (up to 32) in the called address extension. This address length indicator is binary coded, and bit 1 is the low order bit. Bits 8 and 7 **of** this octet are set to zero.

The following octets (up to 16) contain the called address extension.

Each digit of an address is coded in a semi-octet in binary coded decimal, where bit 5 or 1 is the low order bit of the digit.

Starting from the high-order digit, the address is coded in octet 2 and consecutive octets of the facility parameter field with two digits paer octet. In each octet, the higher order digit is coded in bits 8, 7, 6, and 5.

When necessary, the facility parameter field shall be rounded up to an integral number of octets by **inserting** zeros in bits 4, 3, 2, and 1 of the last octet of the field.

AX.25 marker definitions

In order to make clear the various markers that might be in the facility field, they are listed below. Once again these markers are used to separate the various types of facilities that might appear in call generation packets.

Facility marker for calling network facilities

This marker signifies facilities following it are to be provided by the calling DTE network.

```
bits:    8 7 6 5 4 3 2 1
octetl:  0 0 0 0 0 0 0 0
octet2:  0 0 0 0 0 0 0 0
```

Facility marker for called DTE network facilities

This marker signifies facilities following it are to be provided by the called DTE network.

```
bits:    8 7 6 5 4 3 2 1
octetl:  0 0 0 0 0 0 0 0
octet2:  1 1 1 1 1 1 1 1
```

Facility marker for CCITT-specified facilities

This marker signifies facilities following it are CCITT-specified facilities.

```
bits:    8 7 6 5 4 3 2 1
octetl:  0 0 0 0 0 0 0 0
octet2:  0 0 0 0 1 1 1 1
```

Facility marker for amateur network facilities

This marker signifies facilities following it are amateur radio network provided facilities.

```
bits:    8 7 6 5 4 3 2 1
octetl:  0 0 0 0 0 0 0 0
octet2:  1 1 1 1 1 1 1 0
```

Note: The amateur facility marker has been changed, since the CCITT has added a marker using the original code that the AX.25 draft committee used. The choice of 11111110 is **being** made in hopes that the CCITT will stay away from this code, since the code 11111111 has been used.

Coding of amateur addresses in AX.25

The following is a recommendation on coding of the calling and called DTE address fields and using the calling and called extension facilities in an amateur AX.25 network.

Coding of the DTE address fields

If the actual DTE addresses are conveyed in the newly created calling and called address extension facilities, this leaves the DTE calling and DTE called address fields available for other uses. One use for these fields might be to convey geographical location information of the DTEs involved, which might help call routing decisions.

If we leave the first two octets for the DNIC number (four digits coded per AX.25 section 6.2), this leaves room for up to 5 octets (10 digits) of additional information. One recommendation would be to use the VHF grid location system.

One of the problems with the VHF grid system is that it uses alpha characters in the first two characters, and numeric characters of the last two characters of the coarse location, and two more alpha caharacters in the two additional characters used in the fine grid location system. Since AX.25 specifies binary coded decimal format digits in the address fields, ASCII characters could create invalid DTE addresses.

A suggestion to avoid this problem is to break up the alpha characters into two portions, each representable in binary coded decimal format. If an ASCII character (upper case alpha, or numeric only) is divided so that bits 1, 2, and 3 are conveyed in the low order digit, and bits 4, 5, and 6 are conveyed in the high order digit, an ASCII character could be represented in one octet, while still keeping to the letter of X.25.

Using this technique, the first two octets would convey the DNIC number, the third octet would convey the first alpha character of the VHF grid system, the fourth octet would have the second alpha character, the fifth octet would have both digits of the grid system identifier. The sixth and seventh octets would carry the fine resolution alpha characters of the grid information.

Address extension field coding

The addition of the calling and called address extension facilities has allowed a re-thinking of amateur address coding. As the description of these facilities above shows, the CCITT is still considering addressing information to be numeric only. The United States contigent is hoping that this linitation can be eliminated, however.

Anticipating this loosening-up of the restriction, I recommend that the extended address coding consist of the amateur callsign of the station involved, consisting of the six upper-case alpha or numeric characters, followed by an additional octet carrying a substation identification number. this substation identification number should be five bits long, binary coded, and reside in bits 5, 4, 3, 2, and 1 of the seventh octet of the address field. Bits 8, 7, and 6 are reserved at this time, and set to zero.

This coding scheme will allow the amateur callsign to be used as a unique station identifier, just as it is in Level 2 of AX.25

An Enhanced Terminal Node Controller

by Lyle Johnson, WA7GXD
President
Tucson Amateur Packet Radio
PO Box 22888
Tucson AZ 85734

ABSTRACT

This paper describes changes made in the Terminal Node Controller (TNC) developed by Tucson Amateur Packet Radio (TAPR) as a result of extensive field evaluation during the TAPR Beta test.

BACKGROUND

Tucson Amateur Packet Radio is a non-profit Research and Development group with over 700 members worldwide. During 1981-1982, a complete, self-contained Terminal Node Controller was developed. During the period of January, 1983, through June, 1983, over 170 preassembled TNCs were placed in more than 19 sites worldwide for testing and evaluation.

The testing revealed many things beyond the evaluation of the TNC: primary among this list is the finding that non-technically oriented Amateurs were both interested in Packet radio as a mode and capable of placing a Packet station in operation.

Another important observation was that a volunteer group would quickly reach burnout if the participants were required to assemble and test each and every TNC prior to shipment. This observation, along with tightened FCC type classifications under Part 15 for manufactured equipment, led TAPR to the first major change in the Beta TNC for general distribution -- the new TNC would be in the form of a kit.

THE BETA TNC

TAPR's first generally distributed TNC design has become known as the Beta Board. This device was based on the 6809 microprocessor, included 6k-bytes of RAM, 24k-bytes of EPROM, 64 bytes of non-volatile memory (NOVRAM), a Western Digital HDLC controller, serial and parallel ports, an on-board 1200-baud radio modem with watchdog timer, built-in regulated AC power supply and a generous wire-wrap area for the experimenter.

Software was written in the PASCAL language for high-level functions, with low-level routines written in 6809 Assembler. The software design was accomplished by a team scattered between Tucson and the Los Angeles area, while the hardware design was effected in Tucson.

A comprehensive manual was written and distributed with the Beta Board; it contained nearly 200 pages of descriptions, operating procedures, interfacing data, protocol definitions and general Packet radio background information.

THE KIT TNC: DESIGN EVOLUTION

The response of the Amateur community to the Beta Board was overwhelmingly positive: nonetheless, there were some minor difficulties reported. In addition, TAPR wanted to increase the flexibility of the TNC to more easily accommodate the anticipated users of PACSAT, HF, METSCAT, OSCAR 10 and others. Finally, the unit had to be designed in such a way as to allow assembly and calibration by the less technically oriented members of the Amateur fraternity to help encourage the growth of the mode through on-the-air application.

A few users found the non-polarized connectors used on the Beta Boards to be a problem. Worse, the radio input/output (I/O) and power transformer disconnect used identical connectors. The solution to this problem meant a different form factor for the board: this provided the opportunity to include subtle changes to the TNC's hardware design since a completely new printed circuit (PC) board would have to be laid out.

Field feedback coupled with local experience on the part of the software design team led to suggestions for improved performance and an even more friendly user interface. Naturally, this meant an increase in the standard program code storage capacity (from 24k-bytes to 32k-bytes)!

The documentation would require a complete rewrite. The operating section would have to reflect new commands and capabilities. The circuit descriptions would have to be revised. And a detailed assembly manual would have to be written and field-tested.

The opportunities roughly defined, it was time to go to work! During the next few months of intensive effort, the design team kept true to the premise that

> "the sooner you get behind in a project, the more time you have to catch up!"

HARDWARE ENHANCEMENTS

The TAPR kit TNC offers 8k-bytes of RAM (up from 6k-bytes on the Beta Board) as standard. The Program storage has been increased from 24k-bytes to 32k-bytes. In addition, through the use of an 8k-byte single chip static RAM chip (TAPR was one of the very first US corporations to use this IC), an empty memory socket is provided with jumper options to allow the use of any JEDEC "bytewide" memory IC from 2k- through 32k-bytes in capacity. The memory decoding circuitry is pre-programmed to accommodate a 16k-byte part in this site. Thus, the TNC can utilize up to 56k-bytes of memory without any modification whatsoever!

Nonvolatile RAM (NOVRAM) has been increased from 32 bytes to two "banks" of 64 bytes each. This non-battery backed memory technology allows the TNC to store callsign, terminal parameters, radio link data and similar information indefinitely without requiring the operator to manually enter them each time the unit is powered up, or forcing the operator to utilize very conservative default values for the link that are not optimum for his particular local area network.

To simplify interfacing the serial I/O port to ((three-wire" RS-232 devices, default pullup resistors have been included to eliminate the need for jumpers, yet provide hardware handshaking for those attached devices that can support CTS/RTS and DTR/DSR protocol. As with the Beta Board, true RS-232-C levels are implemented on the serial port, although the TNC can communicate with devices that utilize TTL output levels.

The power supply circuitry utilizes 3 amp diodes in the 5-volt supply and the electrolytic filter capacitors are much further from the heatsink to slow the drying of the electrolyte that normally occurs with time. The negative voltage regulators are more heavily bypassed to suppress any tendency towards oscillation. Finally, a separate 5-volt regulator and ground return path is provided to the on-board modem for more complete noise rejection. The 5-volt regulator IC and heatsink are wired in such a way as to allow easy relocation off-board in the event a cabinet is used while allowing on-board operation if the TNC is not enclosed. A Molex connector is provided for the power transformer disconnect: a redesigned power transformer is included with multiple line voltage taps for best efficiency.

The clock oscillator frequency has been doubled and a jumper provided to allow operation of the digital circuitry at a faster rate in the event high-speed packet operation (greater than about **9600** bauds) becomes commonplace.

Link status is output on the parallel port in the default condition: this allows easier interfacing to "host" computers that may have one or more slave TNCs attached for bulletin board and other services.

A standard DB-25S connector is provided on the TNC for the serial portandthe TNC is configured as "Data Communications Equipment'" (DCE) -- it appears as a standard modem to an attached terminal or computer.

A DB-25P connector is used to interface to the parallel port, while a DE-% connector is included to attach a radio to the TNC. The pinout of the DE-9 is such that, if ribbon cable is used and attached to all nine pins (as would occur if an insulation displacement connector were used), alternate leads will be grounded for good signal separation and shieldinq.

The TNC was designed with a cabinet -in mind. In addition to the heatsink considerations mentioned above, there is a complete "front panel" disconnect on the unit. A 16-pin IC socket located immediately behind the front panel of the cabinet repeats all on-board LED functions as wel l as all DIPswitch connections. The LEDs have their own current limiting resistors.

LED monitored functions have been increased from those on the Beta Board. In addition to received audio input level monitoring and transmitted data, there are monitors for data carrier detect (very useful for HF and OSCAR operation), CWID, PTT, HDLC RESET and SPARE (user configurable). The switch functions have been increased and include ROM/NOVRAM default parameters, NOVRAM BANK SELECT (for multiple operator or multiple function stations), NOVRAM DISCONNECT (for rebooting with "non-permanent" parameters) and, of course, RESET.

The area that received the greatest attention in the hardware rework effort was the modem. This is perhaps the most critical part of the entire TNC hardware.

With flexibility and performance as the bywords, the modem design -was picked apart by several people and extensively tested. The results of these investigations were consolidated and incorporated in the new TNC design.

Careful rework of the power supply and power distribution on the hoard resulted in a reduced noise floor for the modem. Increased physical separation of the modulator and demodulator reduced crosstalk tendencies. Post filtering of the MF-10 switched capacitor filter by an op-amp removed high frequency switching transients from the input to the XR2211 PLL demodulator. The demodulator can be completely reconfigured (for 300 baud HF work, for example) by swapping a 16-pin

DIP header plugged into a standard IC socket. This encourages experimentation with other shifts and data rates for optimizing communications over a given path.

As with the Beta Board, the MF-10 filter is defined via a plug-in DIP header containing only resistors. Similarly, the XR2206 phase-coherent FSK modulator is configured via a 14-pin DIP plug-in header.

One of the (apparently) least understood causes of problems with Packet signals is due to overdrive of the microphone amplifier circuitry in the associated radio. The Beta Board used the output of the '2206 modem (buffered by an op-amp with unity gain) to feed the radio directly. This resulted in a rather critical setting of the audio output adjustment trimmer near the low end of its rotation. The new TNC includes a 30:1 attenuator to provide a greater usable adjustment range and minimize the possibility of overdrive to the attached radio.

The radio PTT circuitry is now based on a power FET. Unlike the Darlington transistor keying circuit used in the Beta Board, with its 0.6 volt minimum level when "on," the new TNC has an effective "on" resistance in the keying circuit of about 4 ohms. The "off" voltage can be nearly 36 volts as well. TAPR has not received a report to date of anyone having trouble keying any common radio with this new circuit: several late model radios would not tolerate the standard Beta circuit.

The watchdog timer can be reconfigured via the modulator header. This allows easier interfacing with HF and other radios that typically use lower data rates on the air. The default timeout was increased from about 15 seconds to about 1 minute for similar reasons.

The calibration circuitry for the modem was revised. The XR2211 has an unusual waveform that many commercial frequency counters won't accept: the Beta Board waveshaping circuit only worked for about 60% of the TNCs. The kit TNC includes a Schmitt trigger in the wave shaping circuit that drives the on-board counter: this has proven effective in every case that the author is aware of.

The CWID is now done via FSK rather than tone on-off keying. This prevents a listening TNC from mistaking the pause between letters in a call sign for a free channel and "colliding" with the remainder of the CWID. Further, the TNC suppresses the audio output from the '2206 modulator when it is not attempting to key the transmitter: this allows the user to use voice override without disconnecting the TNC from the radio.

Of course, no matter how good the modem is, others will want to try their own designs. This being a goal of TAPR's, a complete modem disconnect is provided on the kit TNC. Utilizing a standard IDC 20-pin polarized connector, it is a simple matter to reconfigure the TNC to use an external modem. This is desirable when experimenting with radio link data rates in excess of 1200 baud, or when using a scheme other than (A)FSK for modulation.

SOFTWARE ENHANCEMENTS

The added memory space for program storage allowed the already friendly user-interface to become friendlier.

Several commands now have more flexible arguments: for example, the DISPLAY command, which used to dump all user-alterable parameters, now may optionally show only those parameters that relate to a specific use, such as terminal characteristics.

TAPR took the liberty to allow multiple digipeaters to be specified in AX.25 mode. The Beta software allowed up to three "hops" while the new kit software allows up to eight! While digipeaters are a bit of a kludge to a "pure" level two, they have proven invaluable during this early phase in the development of packet radio and the multiple hop concept has been invaluable in many areas that otherwise would be cut off from "local" activity.

Link status is now output on the parallel port. This was provided in response to numerous requests from users who wanted to interface bulletin boards and other "host" computer applications.

Baud rates for both the terminal and radio ports are now specified by the data rate (e.g, 1200) rather than by a table lookup value (e.g., 8). Similar human factors improvements have been made in specifying many other parameters.

The suppressable CWID may now include text instead of simply the station call sign.

Beacon text and non-connected-mode packet transmissions may now specify up to eight digipeaters in their output routing.

To provide greater compatibility with users of the VADCG board, special options were included to allow the sending of CR and LF characters as well the methods of receiving them.

In addition to the standard RTTY-like CONVERSE mode, a full "transparent" mode of operation, proven on the Beta Boards, is retained for such tasks as computer-to-computer file transfers.

A special full-duplex mode (the TNC always operates in a full-duplex mode on both the radio and serial I/O ports) has been included for use on OSCAR 10 and other very noisy environments.

Support for the TAPR EPROM programming adapter is included in the software. This allows duplicating the TNC software for field updates and the like. Using the INTEL "intelligent" programming algorithm, 2764s are typically programmed in 1-1/2 minutes and 27128s in under 3 minutes.

The above serve as examples of the extensive software effort associated with the release of the new TNC kit -- there are many more changes, some very subtle, which contribute to the ease of use and general acceptance of the device.

Of course, with the exception of the EPROM programmer support, a set of EPROMs that incorporate the full "Version 3" software enhancements is available for use on unmodified Beta Boards -- modified Beta Boards can use kit software.

DOCUMENTATION

The TNC manual was extensively revised to include full operating details for the new software. A tutorial-like "front-end" was added to the operations section for the first-time user who is not in an area where there is other local Packet activity*

The largest single "new" effort in the manual, however, was in writing the assembly instructions. The instructions had to be sufficiently detailed to allow a relatively inexperienced constructor to assemble, calibrate and verify the operation of the TNC.

While not often considered as documentation, the PC board had to be silk screened in a non-ambiguous manner and keyed to the manual in such a way as to reinforce the accuracy of assembly and, hopefully, assist in locating assembly errors and later troubleshooting during the useful life of the TNC.

The manual organization was carefully considered and a looseleaf notebook format with tabbed index dividers was developed. The result is an easy-to-use reference for Packet station operation along with a clean appearance.

KIT TEST

In the late summer of 1983, the kit TNC was ready to be tested. Two "Rev 1" TNCs were built and evaluated. After numerous delays (if anything can go wrong, it will), twenty "Rev 2" test kits were sent to carefully selected participants in early October. The testers were asked to build the kits, evaluate the instructions (the manual was very preliminary, as was the software) and give a detailed report on the overall "kit-ability" of the unit.

The response was gratifying. Every kit worked! Valuable insight was gained into the assembly process, the directions were revised and the graphics taken to a commercial firm for finalization.

PRELIMINARY FIELD RESULTS

Over the period from November, 1983 through early March, 1984, TAPR shipped over 500 kits. None have been returned for repair, although several dedicated packeteers in a few locations have assisted other locals in getting some kits up and running.

Special MF-10 filter resistor packs have been developed and used successfully for routine HF and OSCAR-10 communications.

Packet QSOs with TNCs of other manufacture have been made regularly, the only difficulties being in the non-uniform implementation of the "Poll/Final" bit of the AX.25 protocol specification.

Various bulletin board systems have been successfully interfaced to the TNC. Radios of many types and computers/terminals of various manufacture have been connected with minimal difficulty.

Experiments with the on-board modem have resulted in solid copy over marginal paths at 1200 baud; 1800 baud operation has been regularly achieved over very good paths.

CONCLUSIONS

The development of the TAPR TNC kit was a very challenging experience to those involved. The TNC itself has been widely accepted in the Amateur community; the fact that it is a complex piece of equipment offered only in kit form has not seemed to restrict that acceptance.

An extensive refinement of the Beta Board tested during 1983, the TNC represents the unselfish efforts of hundreds of Amateurs worldwide to help usher in an era of error-free digital communications for the Amateur community at large.

REFERENCES

[1] Henderson, David L., KD4NL, "Design Decisons for the TAPR TNC Link Level," Proceedings of the Second ARRL Amateur Radio Computer Networking Conference, 1983.

[2] Johnson, Lyle V., WA7GXD, "Unique Features of the TAPR TNC," Proceedings Of the Second ARRL Amateur Radio Conputer Networking Conference, 1983.

[3] Morrison, Margaret, KV7D, "Real-Time Low-Level Software on the TAPR TNC," Proceedings of the Second .ARRL Amateur Radio Computer Networking Conference, 1983.

[4] Morrison, Margaret, KV7D, and Morrison, Dan, KV7B, "Designing the TAPR TNC

Audio Input Filter," Proceedings of the Second ARRL Amateur Radio Computer Networking Conference, 1983.

[5] Price, Harold E., NK6K, "Multi-Use Design Considerations for the TAPR TNC," Proceedings of the Second ARRL Amateur Radio Computer Networking Conference, 1983.

[6] Tucson Amateur Packet Radio, Packet Radio System Beta Test, February, 1983.

[7] Tucson Amateur Packet Radio, Packet Radio Terminal Node Controller System Manual, First Edition, November, 1983.

Some Thoughts on AX.25 Level Two

by Lyle Johnson, WA7GXD
President
Tucson Amateur Packet Radio
PO Box 22888
Tucson AZ 85734

ABSTRACT

Comparisons are made between commercial packet-switching applications and the unique Amateur radio environment. Suggestions for enhancing the AX.25 Level Two protocol are given.

BACKGROUND

In October, 1982, a special meeting was held in conjunction with the AMSAT Annual Meeting to define a Level Two protocol. Representatives from many Packet groups were present, and adopted a modified version of the AMRAD-sponsored AX.25 Level Two protocol.

Since that time, AX.25 has become the de facto standard Level Two protocol in the United States and many other countries.

Tucson Amateur Packet Radio (TAPR) implemented this new protocol (with a few notable extensions) in December, 1982, on its then-current "Beta" Terminal Node Controller. These devices saw widespread distribution beginning in January, 1983.

Since that time, over 700 TAPR TNCs have been placed in the field and the extensions have had widespread acceptance. With experience have come requests for certain other changes to the protocol -- these requests form the basis of this paper.

COMMERCIAL APPLICATION

X.25 (the basis for AX.25) is used in commercial packet-switching networks. There are specific features to this protocol that allow for such things as assessing connection charges and the like, but a primary philosophical factor reflected in the protocol is that of "point-to-point" connections.

To expand on this thought, X.25 assumes that the "terminal node," or user, is connecting to a "host," or master, node. All communications to and from the user go through this host. This, of course, makes it easy for the host to log connect time and otherwise supervise the network so each user gets his bill on time!

Another feature allowed in X.25 is the so-called "balanced mode," where two nodes are connected as equals; there is no master/slave connotation. This is the mode that has been adapted to Amateur use.

Balanced made has two outstanding features that are particularly useful for radio Amateurs.

First, every station has the same privileges. This is necessary in a "controlled anarchy" environment such as Amateur radio. Any station can initiate a connection (QSO) -- and any connected station can initiate a disconnect.

Second, by not requiring any master station, the system is very robust. Failure of any particular node does not cause the network to fail.

Amateur Needs

Amateur radio has some specific needs, however, that are not addressed by X.25. One of these needs relates to the address field: AX.25 provides a useful solution by encoding the Amateur call sign in the address field of the header and allowing up to16 stations per Amateur call via the Secondary Station ID (SSID) portion of the address.

Another need is related to the geographic area that a "local area" network may have to encompass. What happens if your station is behind ahillandyou cannot access the local Packet bulletin board system?

AX.25 provides for a "digipeater." This is an intermediate station that can be specified by the initiator of a connection to act as a relay between the two end stations. While this application violates "pure" level two protocol, it satisfies a real need.

When TAPR was implementing AX.25 for the first time, the software team (Margaret Morrison, KV7D, David Henderson, KD4NL, and Harold Price, NK6K) saw a need for multiple digipeaters. Since AX.25 didn't allow for this, they decided upon an AX.25 compatible scheme.

Basically, three digipeaters were allowed to be specified in the "VIA" argument in a connect request. Each station that received the packet scanned the digipeat field and looked for the first "I haven't been digipeated yet" bit that wasn't toggled to the "I just digipeated this frame" state. It then toggled the bit and trans-

mitted the resultant packet. The end recipient simply reversed the order of the digipeater list, cleared the digipeated bits and sent the reply.

Since digipeating allows for end-to-end ACKs only, the NAK being implicit, some mechanism had to be found to give digipeated traffic priority in a net. This was solved via the DWAIT parameter. Essentially, every packet transmission that is not a digipeated packet waits the usual random backoff time, but always waits a minimum of DWAIT * 40 mSec. Digipeated packets do not wait this delay; they have priority on the channel.

This feature was found to be very useful in such areas as Los Angeles-San Diego and the greater St. Louis area.

When the TAPR kit TNC was developed, a new software release was simultaneously released. The Version 3 software allows up to eight digipeaters to be specified, and also allows the use of digipeaters in the beacon and unconnected modes of operation.

Since this extension is a violation of the AX.25 protocol as adopted at the AMSAT meeting, the TAPR implementation allows for totally compatible operation as long as not more than one digipeater is specified by the user. It is hoped that other packet groups will recognize the benefits of allowing multiple digipeaters, and at such time as an AX.25 Level Two protocol review meeting is held with participation by interested Packet groups, TAPR will formally propose that these extensions be incorporated in the protocol.

While on the subject of implemented extensions to AX.25 Level Two, TAPR has extended the use of the Disconnected Mode (DM) frame.

AX.25 specifies that this frame will be sent only when the addressed station is in the disconnected mode and receives a frame other than a connect request (SABM).

The TAPR TNC has a command thatallowsthe operator of the station to set a CONnect OK (CONOK) flag to OFF, thus inhibiting his TNC from being connected to. This allows the operator to listen on the channel without having to "talk" to anyone. Under these conditions, a SABM frame will be responded to with a DM frame.

The other non-standard sending of a DM frame occurs when the destination TNC is already connected to another station.

The station requesting the connection, if in CONVERSation mode (not TRANSparent mode), will get a message stating

*** <call> busy

when a DM frame is received. Likewise, the station sending the DM frame will display

*** connect request from <call>

to alert him that a(nother) station wishes to connect.

OTHER EXTENSION

There are two other cases that arise in common Amateur practice that the author believes should be addressed at "Level Two" in Amateur Packet radio.

The first is the case of multiple simultaneous connections. This occurs when more than one station desires to use the services of another station.

A "sort of" case of this occurs when one station is in a good location and becomes a digipeater used by other stations in the local area. While no connection exists to a digipeater (only through it), the station so used is an illustrative example of of multiple connections.

One of Packet's widely touted benefits is its time domain multiplexing (TDM) on a given channel. This allows multiple QSOs to take place, increasing channel utilization. However, when a Packet station connects to the local Packet bulletin board, it becomes apparent that the bulletin board is being underutilized. Other station must wait in line for the first station to disconnect before the next one can connect. Meanwhile, the BBS is often standing idly by while the connected user browses through his mail or digests something just read.

If multiple connections were allowed, many users could potentially access the BBS at the same (apparent) time.

Please note that this is a question of implementation of AX.25 Level Two -- nothing in the protocol prohibits multiple connections. The upcoming Version 4 software release for the TAPR TNC will allow such multiple connections.

One major difference between Amateur operation and commercial practice is in the use of roundtables. This is a mode of operation where there are several stations that are engaged in a multi-way conversation.

Such operation is very useful when one wants inputs from a number of others on a particular subject, or when a traffic net has items of general interest (a swap net comes to mind as a typical example).

AX.25 Level Two does not allow for this mode of connection. While the next layer(s) of protocol will undoubtedly allow some semblance of this kind of operation, it will probably be dependent on some sort of "master@' linking station. This may reduce the robustness of the local system, which could be especially critical in times of local emergency traffic handling.

It is the author's belief that such operation is totally feasible within the Level Two environment by simply making use of the two "reserved" bits in the seventh octetofeachcall in the address field.

While this is not a formal proposal, the idea is as follows.

[1] The use of up to ten call signs is permitted in the address field (in the same manner as implemented in the TAPR-extended digipeater string). This allows up to nine destination stations in the multi-way connection.

[2] If the two bits marked "RR" in the seventh octet of the call are set to a "11", the call is treated like a digipeater -- this allows digipeaters in the case of certain stations in the multi-way connect, but reduces the number of destination stations by the number of digipeaters specified.

[3] If the 6th bit (counting from 0) in the seventh octet is a "Ø", such that the field marked "RR" is "1Ø", the station is treated as a destination station in the multi-way connect. Such a station would scan the previous addresses to see if this framewastohavebeen sent via a digipeater, and if so, if it in fact has been digipeated. The station would then continue the scan to see if it was requested as a digipeater for some other destination station in the multi-way connect.

[4] If the station is a destination station, it would read the control byte and act accordingly.

This mechanism allows a single packet transmission to be explicitly sent to multiple destinations, avoiding the inefficiencies that would result from a channel bandwidth utilization standpoint if the sending station had to use the multiple connection approach and send a packet to each destination individually.

The nextproblemto solveisthemanner in which ACKs are handled.

Each destination station would only have to send an ACK to the station originating the packet in question. Thus, a non-multi-way packet would be sent, the digipeat field being assembled by reading the address list backwards from this destination station to the first encountered non-digipeater.

A variation of the TAPR TNC DWAIT parameter would be used, wherein the station initiating the ACK would hold off for some number of milliseconds times his position in the address field. This would avoid collisions in most cases, while streamlining the ACK process -- a sort of slotted ACK.

To clarify the above, assume a station sent the following address field (an * indicates a digipeater, a # indicates a destination station):

WA7GXD |NØADI *|N7CL #|NK6K #|NØADI #

In this case, WA7GXD is sending a packet to N7CL via NØADI, to NK6K directly and to NØADI directly.

N7CL would ACK via NØADI; NK6K and NØADI would ACK directly, with N7CL sending the first ACK, followed closely by NK6K and NØADI.

If WA7GXD did not correctly receive the ACK from NK6K, the packet transmission would be repeated, but either (a) would only be sent to NK6K (Non-ACKers only) or (b) would be sent to all stations again, but the already-ACKed stations would ignore the packet because the N(R) and/or N(S) counters would not have been updated by WA7GXD.

This informal proposal is not being presented with the idea that it is the best solution to a multi-way connection at Level Two: rather it is suggested as a possible, compatible means of achieving this end.

CONCLUSION

AX.25 Level Two has been proven as a workable protocol in the Amateur Packet radio environment. With suitable extensions, based on field feedback from active Packet users, it can be made even more suitable for long term usage.

Some extensions have been implemented and tested in the field for an extended period of time: these extensions have been outlined.

The need for as-yet unimplemented extensions to allow multiple and multi-way connections has been pointed out and a possible approach for multi-way connections suggested.

The OSCAR-11 Packet Experiment

by Lyle Johnson, WA7GXD
PACSAT Team Member
c/o Tucson Amateur Packet Radio
PO Box 22888
Tucson AZ 85734

ABSTRACT

UoSAT/OSCAR-11 contains a digital store-and-forward communications facility. This paper presents some background on the development and implementation of this experiment as well as describing its design.

BACKGROUND: PACSAT

The Radio Amateur Satellite Corporation (AMSAT), in conjunction with Volunteers in Technical Assistance (VITA), is currently developing a Packet radio "flying mailbox", dubbed PACSAT (for PACket SATellite). This satellite is projected to contain two identical digital store-and-forward communications facilities, each containing 2-megabytes of storage capacity with multiple uplink channels in the 435-438 MHz satellite sub-band and downlinks in the 2-meter satellite sub-band.

During a design review conference held in Boston during the weekend of 28 July 1983, representatives from the University of Surrey (creators of UoSAT/OSCAR-9) indicated a potential launch opportunity for an Amateur satellite with the LANDSAT D' mission, scheduled for February, 1984. While this launch was far too soon for PACSAT, which will incorporate many new design concepts, it was viewed as a candidate for flying a communications experiment based on some of the new technology expected to be applied in PACSAT.

The opportunity not being officially confirmed by NASA, and the development schedule being brief to the point of absurdity, no public announcement was made of the possibility of an early 1984 Amateur digital communications satellite after the meeting.

UoSAT/OSCAR-11: THE OPPORTUNITIES

The team at the University of Surrey, under the **direction of Dr. Martin Sweeting,** saw the potential launch of UoSAT-B as a chance to further their experiments on gravity gradient stabilization, thwarted by a recalcitrant boom on UoSAT/OSCAR-9, as well as provide another opportunity to explore the near-earth region of space with various radiation detectors, particle detectors and the like. It would also afford an opportunity to gain further experience with **satellite** construction.

The PACSAT digital design groups viewed the launch opportunity as a chance to prove certain concepts envisaged for PACSAT, as well as a chance for "calibration" in satellite design. The satellite would provide a proving ground for protocol development and experimentation for the PACSAT mission. In addition, it would give some members of the design team a chance for early hands-on satellite experience prior to the actual launch of PACSAT in 1986. Finally, some members of the team saw UoSAT-B as an opportunity to try new battery management techniques in an effort to lengthen the service life of "Phase Two" (low orbit, long-life) Amateur satellites.

VITA viewed the opportunity as **a** way to establish "proof of concept" by actual field trials during the more ambitious PACSAT design phase.

DIFFERENCES IN OSCAR-11 AND PACSAT

PACSAT will use advanced modulation and access techniques to maximize reliability and minimize bandwidth requirements. It will have four uplink channels per Packet experiment and one downlink; there will be two complete experiments on board. Each channel is expected to support a data rate of **9600** bits-per-second (bps). Each experiment will contain 2-megabytes of data storage memory. There will be multiple microprocessors in each experiment. HDLC will be used; an extended AX.25 **style** protocol will be implemented and the satellite will be generally available to all Amateurs.

OSCAR-11, on the other hand, contains only a single digital communications experiment **(DCE)**. **It** has 126-k of total RAM capacity, 16-k of which is error-detecting-and-correcting (EDAC) memory for program storage, system stack and the like. The limited uplink and downlink resources mean that the DCE must time-share with other spacecraft use of the beacons. OSCAR-11 is a general research satellite: the DCE is a secondary experiment, not the primary one. Modulation is simple AFSK of an F3 signal: the data rate is limited to 1200 baud. The data encoding technique uses simple UART-compatible asynchronous characters instead of the more complex (and more efficient) HDLC formats in common Amateur Packet use.

Further, the DCE **is a** proof-of-concept

experiment to prove (or disprove) the viability of certain recent technological advances for use in low earth orbit in a limited radiation environment. Hence, much of the task of the system is to exercise and report on the state of the memory devices, current consumption, changes in operational behavior and the like.

Nonetheless, OSCAR-11, through the DCE facility, provides a unique opportunity for developing non-real-time gateway operations for Packet radio use. Such gateway activity, while limited to only a relative few stations with direct access to the satellite, will provide the opportunity for a relatively large number of users to participate in the usage of OSCAR-11 by sending traffic through the gateway stations.

The remainder of this paper will focus on the design and implementation of the Data Communications Experiment, or DCE, currently flying on UoSAT/OSCAR-11.

DCE: DESIGN OVERVIEW

The PACSAT project design team is spread across the North American continent. The PACSAT groups involved in the DCE are:

APU/CCU (Applications Processor Unit/Channel Communications Unit) - based in Tucson, Arizona. This team is responsible for the computing hardware design for the various PACSAT microprocessors.

Ground Station - based in Dallas, Texas. This group is responsible for the PACSAT ground station design (not to be confused with the satellite command stations).

RAMUNIT (Mass Storage) - based in Ottawa, Ontario. This group is responsible for the 2-megabyte mass storage subsystem design to be used in PACSAT.

SOFTWARE - based in Los Angeles, California. This group is responsible for all software that will run in the processors designed by the APU/CCU group.

Unlike previous AMSAT satellites, PACSAT will depend on the use of high-speed microprocessors. The most sophisticated Amateur satellite to date, AMSAT/OSCAR-10, uses the RCA 1802-series COSMAC microprocessor. This device, while available in a radiation tolerant form, is not fast. However, it will run on 10-volts and is more than adequate for the task of managing the spacecraft's resources. PACSAT must rely on processing power to perform its mission as well as manage the various spacecraft support systems: the 1802 simply isn't powerful enough. In fact, no single microprocessor available in CMOS is powerful enough -- PACSAT will use three microprocessors per Packet experiment, or six total. The spacecraft itself may require the use of yet another!

The microprocessor chosen for PACSAT is the National Semiconductor NSC-800. This is a z-80 work alike device, able to draw on the vast resources of proven Z-80 software design tools. Capable of executing instructions at a rapid rate when operated with a 4 MHz clock, studies by the software design group determined that this processor would be capable of performing the necessary tasks associated with data collection and management projected for the PACSAT mission,

However, AMSAT has no flight experience with the NSC-800, nor with any low-threshold CMOS devices. Thus, the NSC-800 was selected to fly on the DCE in order to evaluate the high-speed, low-threshold CMOS technology of which it is fabricated in the low-earth orbit environment.

The DCE's NSC-800 is buffered, and its address/data bus demultiplexed, via 5-volt high-speed CMOS (HCMOS) devices of the 54HCXXX family.

This, too, represents a departure from the relatively "safe" high-threshold silicon-gate technology traditionally flown in space applications. The nature of radiation damage to MOS semiconductors is such that the switching threshold voltage tends to rise- This is due to trapped charges in the gate oxides that occur when bombarded by particles with the right energy levels. A high-threshold device, operating at 10-volts, has more "headroom" in which to sustain damage before its input switching thresholds approach the Vdd level (10-volts). In the case of PACSAT, and the DCE, higher-speed operation is of paramount importance -- standard CMOS devices simply won't do the job needed. Thus, a higher-risk technology is employed to allow the task to be accomplished at all. The DCE provides a mechanism to evaluate this risk in a real space environment.

The primary DCE memory is composed of CMOS static RAM. There is a 16-k byte block of memory that includes a Hamming code EDAC section. This memory is built around Harris 6564 ICs, a hybrid utilizing 4k by 1 CMOS static RAMs. The E'DAC support logic is fabricated entirely of HCMOS devices. An error counter is mapped into the NSC-800 I/O space to allow the processor to determine the number of errors corrected,

The Integrated Housekeeping Unit (IHU) -- the microcomputer that manages the operation of OSCAR-11 -- is designed around the 1802 and has an EDAC memory built around 4116 DRAMs and traditional 4000 series CMOS logic. This unit also has an error counter, and one of the most important areas of measurement associated with the technology evaluation is the comparison of the two error counters. This information, coupled with the absolute levels of radiation as measured by other detectors on board the spacecraft, will help determine the suitability of CMOS static RAMs for

future satellite packages, including PAC-SAT.

The limited knowledge available (at least, in non-classified form), indicates that DRAM is best for overall power consumption over the projected life of the satellite -mission, while CMOS is better initially. UoSAT/OSCAR-11 will give us valuable data as to the relative performance of these technologies in the particular environment of low earth orbits. If the degradation in performance (as measured by changes in nominal current consumptions when the cpu is performing well-known tasks) is acceptable, CMOS static RAM may offer the opportunity to reduce power consumption, a major consideration in spacecraft systems design. The radiation experiments, along with relative errors-corrected data, may show which technology is more tolerant of the particular low earth orbit environment which OSCAR-11 (and later, PACSAT) inhabits.

Additional RAM is provided in the form of 2k-by-8 "bytewide" CMOS static RAMs. A total of 14k-bytes of Harris military grade 6516 components are used. While there is no hardware EDAC circuitry used, this memory will be used to determine the susceptibility of this technology to radiation induced errors -- again, the EDAC memory counter will provide invaluable data for comparison purposes.

EDAC circuitry is not really useful for bytewide components -- a grazing particle may upset the data in several cells, and this could mean that many bits of a particular byte are corrupted, making a very-wide word memory necessary to detect, much less correct, such errors. Instead, it is envisaged that a software encoding scheme could be implemented to correct these errors, similar to the "fire codes" used to recover from error bursts when reading from a Winchester disk, for example. Error detection will be fairly simple in some experiments when a predictable pattern is written to memory and later read back for comparison.

The PACSAT mass storage mechanism currently being explored is that of arranqing a large amount of high-speed static CMOS RAM in a bank-switched scheme in the upper address space of the NSC-800. High density (8k-bytes per chip), low-threshold CMOS devices were selected for a portion of this memory in the DCE (Hitachi HN6264LP-12s), with lower density RAMs serving to implement other banks (Hitachi military-style 6116s). Bank selection and bus isolation is via 5-volt HCMOS logic of the 54HCXXX family.

Bootstrapping the processor from a cold start uses a scheme with bank selected CMOS fusible link PROMs. EPROMs are unuseable due to their dependence on trapped charges for memory retention. The PROM technology selected has been used in space before, although little data is officially available on their performance.

To help ensure reliability, a pair of PROMs, bank selectable by the spacecraft command decoder, are incorporated in the DCE.

The use of PROM in space is another first for the DCE for AMSAT applications, at least as far as bootstrap memory is concerned. The 1802 has a special mode of operation that allows boostrapping code directly into RAM without any ROM whatsoever1 The NSC-800, on the other hand, does not easily lend itself to such usage (although the Japanese JAS-1 Amateur satellite, scheduled for an early 1986 launch, uses a ROMless NSC-800). This meant that a thoroughly tested, absolutely buqfree bootstrap loader had to be designed, tested and burned into the PROM prior to construction of the flight DCE! While developing such a program may appear simple at first glance, the task is not trivial. The system must be errorless in its loading, function without any RAM whatsoever for the loader (no absolute memory location can be considered safe from failure in space applications) and must work well in a very noisy (error-prone) environment. This task was accomplished by the RAMUNIT group, with assistance from the Software team.

Serial I/O is handled by a pair of 6402 UARTs. These devices were successfully flown in UoSAT/OSCAR-9 and they are compatible with the NSC-800. There are two UARTs and they are multiplexed to enable communication with any receiver, transmitter or serial data stream in the spacecraft (the IHU, for example). Serial I/O may be operated in an interrupt-driven or a polling fashion.

Parallel I/O is handled by a Harris 82C55A. It is allowed to control the serial port multiplexers, read data from the navigation magnetometer, select the UART data rate clocks, determine the cpu clock rate and select the RAMUNIT active bank.

A tap in the clock oscillator/divider is coupled to an interrupt line to allow the NSC-800 to keep track of elapsed time for various internal experiments.

Finally, level-shifters are provided to interface the DCE with the rest of the spacecraft. Command control is asserted over the DCE reset line, bootstrap PROM select, cpu clock rate (0.9 or 1.8 MHz) and DCE power on/off. Telemetry status points indicate the state of all the command lines to the DCE, while telemetry channels provide measurement of current to the RAMUNIT, the CPU and the EDAC memory subsystems.

DCE: DESIGN HISTORY

With the limited time available to implement the DCE, many decisions had to be

made rapidly and implemented in no less timely a fashion1

Investigations were made into memory technologies, component availability and EDAC memory design by the APU/CCU group during the month of August. Meetings were held with various AMSAT engineering people and vital information obtained.

The wire-wrap prototype of the NSC-800 based cpu and memory systems was accomplished during the Tucson floods of September, 1983.

By October, the Ground Station group was busily preparing to lay out the PC boards for the CPU and GMEM (General MEMory) cards. The wire wrap prototypes were made functional in Dallas and the layout work began in earnest.

By the first part of December the artwork was delivered to Tucson and the actual flight PC boards fabricated. These were hand carried to Dallas where the engineering prototypes were constructed. The remaining boards and parts were then sent by air to Ottawa where the actual construction of the flight units was to take place.

Meanwhile, the NiCd cells were procured and sent to Ottawa in October for evaluation, classification, and integration into the spacecraft's battery. The cells were then shipped on to Surrey. The RAMUNIT design was completed and the PC boards laid out and fabricated in Canada.

The group leader in Ottawa was severely burned in a fire about this time: although his recovery was very painful, he doggedly persisted on the project.

Software for the DCE was developed in Ottawa and Los Angeles, with the final burning into PROM carefully controlled by the Ottawa group.

In early January, the flight boards were constructed and integrated into the flight module (three circuit boards crammed into a container only 31 mm thick!) in Ottawa, with full-time participation by the PACSAT project leader.

The flight unit was hand carried from Canada to Surrey, where it was further tested and integrated into the spacecraft. The final hardware bugs were exterminated with coordinated efforts from both sides of the Atlantic.

The flight DCE arrived in the States in mid-February as part of UoSAT-B. Magnetometer calibration was accomplished at Goddard Space Flight Center, and the satellite and support team then travelled to Vandenberg Air Force Base, on the California coast. There, it was met by members of the DCE team, and several days were spent verifying the overall health of the DCE and the spacecraft in general. **All** passed with "flying" colors...

OSCAR-11: A STATUS REPORT

UoSAT-B became UoSAT-2/OSCAR-11 on 1 March 1984 after a textbook launch on a Delta vehicle. Initial telemetry was positive.

On March 2nd, however, an anomaly occurred with the 145.825 MHz beacon transmitter. As of this writing (1Ø March 1984), the spacecraft is silent, its command receiver at 7Ø-cm apparently being blocked by the ailing 2-meter beacon. Teams at Surrey and Los Angeles are actively trying to unravel the puzzle and get the satellite in a functioning mode. With persistence and a little luck, 1984 should herald the activation and use of the first major digital store-and-forward Amateur communication facility in space: this should put emphasis on the development of a workable gateway plan.

DCE: THE PLAYERS

The author wishes to recognize the many volunteers who made the DCE possible. Any omission is purely unintentional.

DCE Project Leader : Harold Price, NK6K

Ottawa Team List (Battery and DCE)

UoS-B Battery Development Team

R. D. (Dick) Atkinson	- VE3JBO
R. B. (Bob) Gillies	- VE3JA
S. J. (Stan) Kazmiruk	- VE3JBA
G. S. (Gord) Scale	- Non Amateur

DCE Software

H. J. (Hugh) Pett	- VE3FLL

DCE RAMUNIT and Construction

R. H. (Ron) Archer	- VE3CNM
G. H. (Grant) Bechtold	- VE3JBF
G. O. (Geoff) Clarke	- VE3JBD
M. S. (Murray) Gold	- VE3KHG
J. M. (John) Henry	- VE2VQ
L. S. (Larry) Kayser	- WA3ZIA
S. J. (Stan) Kazmiruk	- VE3JBA
W. G. (George) Roach	- VE3BNO
G. S. (Gord) Scale	- Non Amateur
D. E. (Dale) Ward	- Non Amateur

U.S. Team

DCE CPU and General Memory Design, Layout, Prototype construction and software.

Dave Cheek	- WA5MWD
Chuck Green	- NØADI
Lyle Johnson	- WA7GXD
Harold Price	- NK6K
Bill Reed	- WDØETZ
Jose Sancho	- WB5YFU
Bob Stricklin	- N5BRG

and of course, AMSAT, VITA and the University of Surrey...

A NEW VANCOUVER PROTOCOL

Douglas Lockhart, VE7APU
953 Odlin Road
Richmond B.C. V6X 1E1
(604) 278-5601

Abstract

This paper describes a new datalink protocol which is being developed by the author in Vancouver and being tested in Toronto. It is designed to replace the previous link level protocol commonly known as the 'Vancouver protocol' and it addresses all the major limitations of that protocol.

Historical Background

In the summer of 1979 in Vancouver I had a protocol in operation in the VADCG TNC which is not well known. It was a protocol designed to talk to a 'Station Node' or central controller which dynamically assigned addresses to each TNC which connected to it. All TNCs communicated to each other through the connection services provided by this station node. The channel was shared using carrier sense multiple access (CSMA) techniques similar to most Amateur packet operation today. Although this system worked quite well and in many respects was more advanced than anything currently in use today we had two major problems which have resulted in it not being used today.

Firstly, the station node used the facilities of my S-100 CP/M system which was being used to develop both the station node and TNC programs. The VADCG did not have the funds or equipment to dedicate to the station node and I was not ready to donate my only computer to the cause. The group tried to assemble equipment which could be dedicated to the station node but were unsuccessful mainly due to funding problems.

Secondly, in the fall of 1979, I was asked by a group in Hamilton, Ontario to write a protocol for them which would allow communication directly from one TNC to another without the need of a station node intermediary. This request was made so that they could use and test the TNCs before they assembled the funds to build their own station node and then switch over to the Station node based software. They had a similar problem to that of the Vancouver group - lack of funds.

And so, by request, I modified the protocol used to communicate to the station node by eliminating the dynamic address assignment system and substituting a fixed address system and made a minimal amount of changes required to provide the requested facilities. Well, as things turned out this 'temporary' protocol became known as the 'Vancouver' protocol and became the 'standard' in North America for a few years. It is still a standard in common use in many areas today. We never have gotten back to the older station node software although it is still in our plans to do so.

When I wrote it, I never intended the program to become a standard for packet radio communications, only for testing TNCs - which it certainly did. It received widespread distribution mainly because it was the only thing available and also because it was capable of doing much more than just test TNCs. In spite of its limitations, it still meets the needs of most users at the present time. This article is written to address those limitations by implementing a new link level protocol which I believe can be used as a base on which to build network and transport level protocols.

A meeting of U.S. (only) packet radio groups was held in October, 1982 which resulted in the adoption of a protocol commonly known as 'AX-25' by most U.S. packet radio groups. (I would have liked an invitation.) AX-25 addresses most of the limitations of the Vancouver protocol but is not a true link level protocol since it concerns itself with several network level functions as well. It is also undergoing changes at the present time and has its own set of problems and limitations. It is my preference to have a protocol which makes a clear separation of link level and network level functions as per the ISO model. As far as standardisation is concerned, when signals with other protocols arrive in the area we hope to be able to provide a gateway interface to handle them.

For purposes of discussion I will refer to the orginal Vancouver protocol as the 'Vancouver protocol' and the new protocol as 'Vancouver Version 2 protocol' or 'V-2' for short.

INTRODUCTION TO V-2

The V-2 protocol is intended to be an efficient datalink level protocol specifically for the Amateur Radio environment but potentially usable in other environments as well. Both full and half-duplex modes are defined. After link establishment, V-2 has only 5 octets of overhead per frame in addition to the standard 3 or 4 octets required for framing. It is strictly a datalink protocol and does not provide any network level functions. The network level protocol will be provided in a Network header which appears only in Information-transfer frames.

All control information for the link and only control information for the link is carried in a Link Header and Link Trailer which appear at the beginning and end of every frame respectively. Only the information not in the link control fields need be passed to higher levels of protocol. This separation of both function and headers allows software for layers to be developed in modular form and permits modification of the code for one layer to have minimal affect on other layers.

This protocol makes no claim or intent at comparison or simulation of the CCITT X-25 standard although it is the author's opinion that V-2 can be compared in the same way as AX-25 can to X-25 with as much similarity or more.

Frame Structure

All transmissions are sent in frames similar to IBM's SDLC and the CCITT's HDLC as well as to AX-25 and the older Vancouver protocol. The frames are encoded in (and decoded from) NRZI (Non Return to Zero Inverted) mode by the HDLC protocol controller chip. The general field format of a V-2 frame is as follows:

SYNC	FLAG	ADDR	CONTROL	INFORMATION	FCS	FLAG

Field Descriptions

Sync field

This field is used for preframe synchronisation. It is either 16 bits of zeroes or sufficient flags for the receiving side to establish bit synchronisation. This field is basically under control of the HDLC protocol controller chip and is transparent to the software.

Flag fields

All frames begin and end with a flag which is the bit sequence 01111110. When sending multiple frames one after the other, the trailing flag of one frame may be used as the leading flag of the next frame. The HDLC protocol controller chip uses a bit

stuffing and deleting technique to ensure that a flag sequence can only appear at the beginning and end of a frame. Flags are automatically inserted by the HDLC protocol controller chip on transmit and are likewise automatically deleted from the data stream before passing to the software. i.e. the flags are transparent to the software.

Address field

This is a four byte address used to identify the link. The old protocol used a one byte address. How it is formed is discussed later. This field is generated and checked by software.

Control field

This is a one byte field. It is used to identify the type of frame and provides information for supervision of the flow of data over the link. This field is generated and processed by software. The format of this byte is discussed later.

Information field

This variable length field is only present in certain types of frames. It must be an integral number of octets. Although the hardware allows a maximum length of over 65,000 bytes, it is recommended that a maximum length of only 200 bytes or less should ever be created for transmission. On the other hand, the system should be capable of handling frames of longer lengths say up to 250 bytes when received from other nodes. This allows room for higher level overhead should it be necessary. If unusually long frames are to be sent, higher level protocols should agree on the size in advance.

FCS (Frame Check Sequence)

This is a two byte field generated as per IS0 standard 3309. This field is automatically generated, checked and removed by the HDLC protocol controller chip so I will not discuss it further. It is used to verify the accuracy of the rest of the data in the frame.

Frame Types

There are three types of frame formats: unnumbered, supervisory, or information transfer format. The frame type is indicated by the value of the low order bit or bits in the control field as seen in storage. (Note that this article refers to the fields as they are seen in storage - not as how they are transmitted by the protocol controller chip) If the low order bit (bit 0) is 0 then it is an information transfer frame. If both bit 0 and bit 1 are 1 then it is an unnumbered frame. Finally, if bit 0 is 1 and bit 1 is 0 then it is a supervisory frame.

Unnumbered-format frames (U-frames) are used during link establishment and termination. They are also used to transfer data when the data is not to be checked as to its location in a sequence of frames. There are a possible 32 types of U-frames but only three of these are used in this protocol.

Supervisory-format frames (S-frames) are used to assist in the transfer of information in that they are used to confirm preceding frames carrying information. The frames of' the supervisory format do not carry information themselves. These frames carry an Nr count which is used to confirm received frames. They also convey ready or busy conditions which assist in coordinating flow control across the link and they are also used to report frame numbering errors (indicating that a numbered information frame was received out of its proper sequence). There are a possibility of 4 types of S-frames but only 3 are defined in this protocol.

Information-transfer format frames (I-frames) actually transfer the data from higher protocol levels across the link. The control field contains send and receive counts (Ns and Nr), which are used to ensure that these frames are received in their proper order (Ns) and to confirm accepted information frames (Nr). The Ns count indicates the number of the information frame within the sequence of information frames transmitted. The Nr count transmitted in a frame is the number (Ns) of the information frame that the node transmitting the Nr expects to receive next.

Frame numbering

A node transmitting numbered I-frames counts each such frame and sends the count with the frame. This count is a sequence number known as Ns. This sequence number is checked by the receiver's datalink layer for missing or duplicated frames.

A node receiving I-frames accepts each numbered frame that it receives (that is error-free and in-sequence) and advances its receive count for each such frame. The receiver count is called Nr. If the received frame is error-free, a receiving station's Nr count is the same as the Ns count that it will receive in the next numbered information frame; that is, a count of one greater than the Ns count of the last frame received. The receiver confirms accepted numbered I-frames by returning its Nr count to the transmitting node.

The Nr count at the receiving station advances when a frame is checked and found to be error-free and in sequence: Nr then becomes the count of the "next-expected" frame and should agree with the next incoming Ns count. If the incoming Ns does not agree with Nr, the frame is out of sequence and Nr does not advance. Out-of-sequence frames are not accepted. The receiver does, however, accept the incoming Nr count (for confirmation purposes) if the out-of-sequence frame is otherwise error free.

The counting capacity for Nr and Ns is 8, using the digits 0 through 7. These counts wrap around; that is, 7 is sequentially followed by 0. up to seven unconfirmed, numbered information frames may be outstanding (transmitted but not confirmed) at the transmitter. All unconfirmed frames must be retained by the transmitter, because it may be necessary to retransmit some or all of them if transmission errors or buffering constraints occur. The reported Nr count is the number of the next frame that the receiver expects to receive, so if, at a checkpoint, it is not the same as the transmitter's next frame (Ns) number, some of the frames already sent must be retransmitted.

The Nr and Ns counts on both ends of the datal ink are initialized to zero during the link establishment phase.

The Poll bit (P-bit) and receive timeout.

The Poll bit is bit 4 in the control field. It is used to force a response from the receiving node when set to 1 (on). Whenever a frame is transmitted with the P-bit on a receive timeout is started by the originating node. This receive timeout (T1) is in the order of 1-3 seconds. This timeout is cancelled by reception of any frame from the other side of the link. If the timeout expires before the expected frame is received then the transmitting node takes corrective action. The number of successive timeouts is counted and reset when a frame is successfully received. If the number of successive receive timeouts exceeds a predetermined level (N1), the next higher level of protocol is notified. The higher level may take other action such as terminating the link.

Note that the receive timeout only proceeds when the data link channel is clear. The timeout is suspended when the data link channel is occupied. On VHF, the datalink channel may be occupied by carriers, voice transmissions, QRM, etc. in addition to the expected data signals. For this reason, it is strongly recommended that CD (Carrier Detect) be tested for receive timeouts rather than DCD (Data Carrier Detect). All VADCG board installations that I know of in Canada are using the squelch line from the VHF transceiver rather than the Data Carrier Detect line from the modem. Some can select both. If the DCD line is used, excessive timeouts will occur and it will be difficult or impossible for the TNC program to determine proper action. In addition, the TNC will transmit on top of other stations using the channel which can lead to unpleasant verbal exchanges. Remember that no Amateur frequencies are exclusively reserved for this type of packet activity.

NODE NAMING AND ADDRESSING

Before describing link addressing it is necessary to understand the network node addressing and naming system in which this protocol is designed to operate. Also, some terms need to be defined.

Node - In this article the word, 'Node' refers to

any entity in the communications system which originates or receives frames.

Node Names

Associated with each node in the network is a 7 character upper case string of ASCII characters. The first six characters of which are the Amateur Radio call sign padded by ASCII blanks (20 Hexadecimal) on the right. The final character is a suffix used to discriminate between multiple nodes which have the same call sign. This suffix will normally be an ASCII blank (20 hexadecimal) and any additional occurrences of the same call sign will be identified by the ASCII numbers 1,2,3, etc. Each node in the network thus has a unique call sign.

For example:
KA6M*** (The * represents a blank)

Or if KA6M has another node operating:
KA6M**1

Node Address

Also associated with every node in the network is a two byte (16-bit) binary address. This address can be derived from the node name by use of a modified cyclic redundancy (CRC) checking algorithm. The algorithm operates on the 7-byte node name handled as a binary number and generates a 2 byte number.

The algorithm does not generate node addresses with FF (hexadecimal) in the first byte because these are reserved for special broadcast functions. Please see Appendix A for details of the algorithm and a sample implementation for an Intel 8080 microprocessor.

The special destination node address of FFFF (Hexadecimal) is used as a general broadcast address and destination addresses beginning with FF are reserved for selective broadcasts and special purposes.

Link Address

The link address field in the frame is four bytes long and is generated by concatenating the 2-byte node addresses of the nodes at either end of the link. Each link actually has two addresses, identifying the two directions that data can flow accross the link. For example, if A and B represent the node addresses of linked nodes then AB and BA are the two addresses of the link between the pair. Address AB represents the link address for data originated by B and destined for A while BA represents the link address for data originated by A and destined for B. The destination node address always comes first.

Use of Selective Receive

This link protocol has been designed to make good use of the selective receive function available on most HDLC/SDLC protocol controllers such as the Intel 8273. The 8273 can be set to pass to the software only frames which have a specific combination of bits in the first byte after the leading flag. In this protocol, this byte is the first byte of the destination node's address. If a node does selective receive only on the first byte of its address, it can eliminate 99.6% (on average) of the link traffic from having to be read into memory buffers and analyzed by software and yet still be able to do multiple link establishments, terminations, etc.

There are a number of modes of operation possible with this protocol using selective receive options as follows:

1. Selective receive only on the first byte of the node's address for using multiple links as above. This would normally be used in anything but the unlinked state but would be used in the unlinked state if it was not desired to do any of the activities in items 2 and 4 below such as a node with a CBBS host.

2. Selective receive only on address FF (Hexadecimal) to monitor broadcast type transmissions and not establish links.

3. Selective receive on address FF and also the first byte of the node's address. (2 selective receive addresses are supported on the 8273) Links can be set up and broadcast messages can be received. Useful in the unlinked state.

4. Receive on all addresses. This is not a selective receive but it is useful for evesdropping on transmissions intended for others or monitoring activity on the channel to put it more politely.

Link States

Data links are set up and discontinued as required by the nodes in the network. They are temporary. The normal life history of a datalink usually progresses through three basic phases or states namely: establishment, information transfer and termination. Although technically not a link state, the situation where a node has no datalinks in any of the above phases is called the 'unlinked' state.

INFORMATION-TRANSFER FRAMES

I (Information) frames are numbered. The Ns count provides for numbering the frame being sent and the Nr provides acknowledgement for the I frames received. When duplex information exchange is in continual process! each node reports its current Ns and/or Nr counts in each I or S frame exchanged. I-frames are only originated by nodes that are in the information-transfer phase.

The expected acknowledgement is an S or I format frame whose Nr count confirms correctly received frames. (S-frames may be interspersed with I format frames, as needed.) An I-frame always has an I-field, in fact, an I-format frame is considered invalid if it does not contain an I-field in this protocol. See figure 1 for the layout of the I-frame control field.

Control Field Type	Control Field Bits 765	4	321	0
I-frame	Nr	P	Ns	0

Where:
Ns is the send sequence number.
Nr is the receive sequence number.
P is the Poll bit.
Figure 1. I-Frame Control field

I-Field Content

The contents of the I-field in an I-frame are not the concern of the datalink protocol. They are determined by higher levels of protocol. The design of the V-2 network level has not been completed and will be the subject of another paper. However, the work on the network level has progressed sufficiently that some advice can be given to network level implementors and testers that will allow different network headers to be identified and to co-exist on the same channel. It is for this purpose that the following Network level information is included here.

The first sub-field in the I-field is called the Network Header. It is a variable length field composed of an even number of bytes. The low order four bits of the first byte indicate the number of words (1 word = 2 bytes) long the the Network Header is. The high order bit indicates if there is another header following the Network Header and the three remaining bits are used to identify different types of Network Headers that may have the same length.

The bit layout of the first byte in the Network Header is as follows:

```
Bitnumber  76543210
           FTTTLLLL
```

Where: F indicates another header follows if 1
T T T indicates the Network Header type
L L L L indicates the length (in words) of the Network Header.

Even if only the link level protocol is in use the following two-byte dummy Network Header should be included as the first subfield in the I-field:

0100 (Hexadecimal)

This will indicate that the header is 2 bytes long and that no other header follows this one.

SUPERVISORY FRAMES

There are three types of supervisory frames defined by this protocol:
RR Receive Ready
RNR Receive Not Ready
REJ Reject

An S (Supervisory) frame must not contain an I-field. It is considered invalid if it does. See figure 2 for the layout of the control field of an S-frame. S-frames are only originated by nodes on links which are in the information-transfer phase.

Control Field Type	Control Field Bits 7 6 5	4	3 2 1 0
RR	Nr	P	0 0 0 1
RNR	Nr	P	0 1 0 1
REJ	Nr	P	1 0 0 1

Figure 2. S-frame Control field.

RR (Receive Ready)

Indicates that the originating node is ready to receive I-frames and acknowledges receipt of numbered I-frames through Nr-I. It must be sent to clear a previous not ready condition (RNR). It can also be sent with the P-bit on to elicit a response from the node at the other end of the link. It is sent with the P-bit off in response to a poll by the other side when no I-frames can be sent.

RNR (Receive Not Ready)

Indicates that the originating node is temporarily not ready to receive I-frames and acknowledges receipt of numbered I-frames through Nr-1. It is usually sent when buffering limitations and other internal restraints in the node are encountered. It can be sent with the P-bit on to elicit a response from the node at the other end of the link. It is sent with the P-bit off in response to a poll by the other side when no I-frames can be sent. This not ready condition must be cleared by the sending of an RR or REJ frame. Note that the node should still be capable of handling S and U frames even in the not ready condition.

REJ (Reject)

Acknowledges receipt of numbered I-frames through Nr-1 and indicates that the originating node is ready to receive I-frames. It requests the retransmission of numbered I-frames starting at the Nr contained in the REJ frame. Its purpose is to speed up recovery of dropped frames when operating full duplex. Note that this frame is only used with full duplex links and should never be sent on half duplex channels. Only nodes intending to operate full duplex need incorporate support for this type of frame. It is optional. REJ frames may be interspersed in the sequence of transmitted frames on full duplex links. The REJ condition is cleared when the requested frame has been correctly received.

UNNUMBERED FRAMES

There are three types of unnumbered format frames supported by this protocol:
XID Exchange station identification
DISC Disconnect
UI Unnumbered information (formerly NSI)

Unnumbered frames are not sequence-checked and do not use Nr or Ns. See figure 3 for the layout of the control field for these frames. Although I am using names and control field formats common to HDLC/SDLC nomenclature, the actual usage of the unnumbered frames in this protocol is different. I tried to use the HDLC name with the closest approximation to what this protocol is doing. I hope this does not cause any confusion. My other alternative was to use the undefined HDLC U-frames and give them my own names.

Note that the previously described I and S-frame usage adheres fairly closely to the HDLC/SDLC standards but there are major divergences in the usage of the unnumbered format frames. This protocol is not intended to be a copy of any other protocol but has been designed pragmatically. When a piece of another protocol fits the need, it has been used but when the need is unique, then unique solutions are designed. This method reduces the amount of development effort. It also helps those who are familiar with other protocols to quickly develop a familiarity with this one. On the other hand, there may be some confusion when divergences with the other protocols occur.

Control Field Type	Control Field Bits 7 6 5	4	3 2 1 0
XID	1 0 1	P	1 1 1 1
DISC	0 1 0	P	0 0 1 1
UI	0 0 0	P	0 0 1 1

Where:
P is the Poll bit.

Figure 3. U-Frame Control field.

XID (Exchange station Identification)

The XID frame is only used during the link establishment phase of the protocol. I was thinking of calling it the 'LINK' frame rather than XID. Before I-frames can flow across the link, information is exchanged between both nodes in the XID frames. The opposite node's XID information is analyzed and a determination is made whether or not the link can be established. If both nodes like what they see, then the link is established and I and S-frames can flow across the link. On the other hand, if one node doesn't like what it sees, then it sends an XID frame to the other with a reason code filled in which indicates what it didn't like and the link is not established. XID frames are only transmitted on links in the establishment phase. The XID frame A, C and I fields are as follows:

AAAACSSSSSSSDDDDDDDPTR

Where:
AAAA is the address field.
C is the control field (XID).
SSSSSSS is the transmitting node's 7-byte name.
DDDDDDD is the destination node's 7-byte name.
P is the protocol level field.
T is the link type field.
R is the reason field.

Protocol level field (P-field)

This is a one byte field in the XID frame indicating the version of protocol being used by the originating node. By testing this field, the receiving node may determine if it can communicate with this type of protocol. At the present time, this field is 01 (Hexadecimal), indicating this is Level 0 of the protocol (bit 0 is on). This version number should be changed whenever a new level is produced which has features or changes which are not compatible with previous levels. This feature has been added in anticipation of future revisions and extensions of the protocol.

The receiving node checks for a match on this field. Note that a node may support multiple levels of protocol in order to communicate with nodes of different levels. For example, a primary station which supports multiple protocols may send an XID saying that it can communicate with levels 0 and 1 of this protocol by sending a P-field of 03 (Hexadecimal) with bits 0 and 1 on. The secondary determines if communication is possible by and'ing its own version(s) with that in the incoming P-field. If the result of this operation is non-zero, then the protocol levels are compatible and the secondary will respond with a P-field indicating which protocol it is using. If the result of this operation gives more than one bit on, then the secondary can choose which of the protocols it wishes to use.

For example:

1. Primary sends P-field of 03 indicating it can support levels 0 and 1.

2. Secondary logically 'ands' the primary's P-field with its own protocol version number of 01 (Only supports level 0). The result is 01.

3. Since the result was non-zero, the protocols are compatible and the Secondary can respond with a P-field value of 01.

On the other hand, if the result is zero, then there is a protocol incompatibility and the secondary will respond with the protocol mismatch bit (another P-bit) set in the Reason field (R-field). The link will not be established.

Link type field (T-field)

The T-field is a one byte field in the XID frame indicating the type of link being established. At present the only choice in link type is either full duplex or half duplex. The byte format is as follows:

```
Bit number  7 6 5 4 3 2 1 0
            X X X X X X X F
```

Where X indicates reserved bits - must be 0
F indicates full duplex if 1, half duplex if 0

This field is checked by the receiving node and if it is capable of handling the type of link indicated it will respond with a matching indication in its own T-field. But if, for example, the primary requests a full duplex link and the secondary only has support for half-duplex, the secondary will respond with the Link type mismatch bit (T-bit) set in the reason field (R-field).

Reason field (R-field)

The reason field is a one-byte field in the XID frame used by the secondary to report to the primary the reason why the link is not being established. If any bits are on in this field, the l-ink was not established. The R-field layout is as follows:

```
Bit number          1 0
     X X X X A R T P
```

Where:
X - reserved bit - must be 0.
A - Address duplication (See Restrictions)
R - Resource limitation
T - Link type field mismatch (see T-field desc.)
P - Protocol mismatch (see P-field description.)

The R-bit being on (1) means that the originating node has a temporary resource shortage which prevents the establishment of the link. The most likely reason being that it has a limitation on the number of simultaneous links that the node can support. Most nodes used in Amateur Radio so far can only support one link at a time, and so, if a node has this link already established when it receives an XID frame from a third party node, it should respond with the R-bit turned on.

Link Establishment

For purposes of the following discussion the node initiating the connection transmits first and is called the 'primary' node. The node being connected to is called the 'secondary' node. This distinction is made because the primary and secondary nodes use some of the fields in a slightly different way. This is a different use of these terms than is found in other protocol documents.

The following chart shows a normal link establishment between the primary node whose address is A and the secondary whose address is B.

```
BA,XID-P --->                  Primary transmits XID
             <--- AB,XID       Secondary transmits XID
BA,I-P ----->                  Primary transmits I
etc.
```

Conflict Resolution

Although unlikely, there is a possibility of the occurence of conflicting requests during the link establishment phase. This is not possible during other link phases. A conflict occurs if two nodes try to initiate a link with the other at the same time but with conflicting requirements. For example, one node wants to establish a half-duplex link while the other wants to establish a full-duplex link or the protocols they want to use are different. Even if both sides take the same recovery action this still does not solve the problem. One side must gain the upper hand. The resolution method chosen is that the node with the higher address gets its way and the node with the lower address must report to its higher level that a link was established but not with the desired specifications due to conflict.

Disconnect frame (DISC)

The DISC frame is only sent during the link termination phase and it is the only type of frame that can be sent on links in the termination phase. I think this frame would better be called 'Unlink' because that is a better description of the function it performs. Connections (in my opinion) are the concern of higher protocol levels - not the datalink level. The DISC frame is only sent after the link has been established and indicates that the originator will no longer send or receive I and S-frames on the link. It is sent with the Poll bit on by the primary node (The node initiating the link termination) and sent with the poll bit off by the receiving secondary node to indicate that it will no longer send I and S-frames on the link.

The format of the DISC frame is the same as that of the XID frame described above except that the usage of the P, T and R fields is different. In the DISC frame the Reason field (R-field) indicates the reason for the disconnect when sent with the Poll bit on by the primary node. If the R-field has no bits turned on this is a normal link termination but if a bit or bits are on in the R-field, then this link is being terminated because of an error. The R-field bits are defined as follows:

```
Bitnumber  7 6 5 4 3 2 1 0
           0 0 0 0 Z Y X W
```

Where:

1. If W is on, the originating node has received an invalid or non-implemented control field.

2. If X is on, the originating node has received a frame with a prohibited I-field.

3. If Y is on, the originating node has received an I-frame with a length greater than the maximum the node can support.

4. If Z is on, the originating node has received a frame with an incongruous Nr.

If any of the bits are on in the R-field, the P-field contains the control field of the frame received which caused the error condition and the T-field contains the following information:

```
Bit number  7 6 5   4   3 2 1   0
              Vr    0     Vs    0
```

Where Vr is the next expected Ns value to be received and Vs was the next Ns value to be sent at the time the error was detected.

The P,T and R fields are undefined (and should be ignored) in a DISC frame with the Poll bit off. Similarily, the P and T-fields are undefined in a DISC frame with a value of 0 in the R-field.

Even though the node names are unnecessary from a technical point of view in the DISC frame, they are included in the Amateur Radio environment to give information to other nodes monitoring the channel as to who the two nodes are who have terminated the link between them. The names are also sent out when the link is being established as well. This is somewhat similar to the Amateur practise of identification at the beginning and end of each QSO.

Unnumbered Information Frame (UI)

The UI-frame is included in this protocol as a method of transmitting information when a link has not been established or for bypassing the normal handling of I-frames when a link is established. One example of such use would be a repeater identifying itself every few minutes - a type of broadcast message. It is included in the protocol because of a need in the Amateur Radio environment for such a function.

Since UI-frames are unnumbered, they should not be acknowledged whether or not they are transmitted with the Poll bit on or off. If one is lost, there is no way of recovering it. No error reporting is done for UI frames and the UI frame should only be passed on to a higher level when the data link is not established, not being established and not being terminated. For use as a type of general broadcast, the link address field in a W-frame should use the special general broadcast address of FFFF (Hexadecimal) as the first two bytes of the link address. If the broadcast is only intended for a

specific area then the first byte should be FF and the second byte should be an area or group number. And if it is intended for a specific node then that node's address can be used in the first two bytes.

Responses to polls

When a node receives a frame with a P-bit on, it must respond. The type of response varies depending on the type of frame that contained the P-bit. The response frame does not have the P-bit on except in the case of the I-frame response where the P-bit is allowed. The responses possible for each type of frame containing a P-bit are shown in Figure 4 following:

Polling Frame Type	Responding frame(s)
I	I or RR or RNR or REJ*
RR	I or RR or RNR
RNR	RR or RNR
XID	XID
DISC	DISC
UI	none
REJ*	I

* = REJ only used in full-duplex
Figure 4. Poll Responses

HALF-DUPLEX PROCEDURES

Half-duplex links assume that only one side transmits at a time. For this reason they can be used on a single two-way communications channel. Half duplex protocols can also be used on independent transmission channels as well but full-duplex would be a more efficient method when these facilities are available. Channel sharing with multiple nodes require half duplex protocols. V-2 provides a half-duplex multipoint protocol. This protocol can be used in a point-to-point environment as well with very little reduction in efficiency compared with a protocol only designed for point-to-point work.

Information transfer

When a node has one or more I-frames to send, it sends them in sequence and in conformity with the requirements described in "Frame numbering" previously. Since I-frames always require a response, the P-bit is turned on in the last frame of the transmitted sequence of I-frames. The exception to this rule occurs when it is necessary to transmit an S-frame with the P-bit on in the same transmission as well. either case, a receive timeout is started at this time as previously described in the section, "The Poll bit and receive timeout". The node then turns the link around and listens for a response. Note that the reception of an I-frame with or without the P-bit on demands a response.

If the receive timeout expires meaning nothing was received, the node will not retransmit the previous frames but transmit an RR or RNR frame with the poll bit on and again wait for a response. This operation repeats itself until something is received from the other side. (The higher level of protocol is notified after every Nl consecutive timeouts.)

When a response is received, the node starts transmitting I-frames again,, beginning with the first unacknowledged I-frame.

Not ready condition handling

When a node becomes not ready to receive I-frames on a link it will send an RNR frame with the poll bit on at the earliest opportunity. If permitted by the other side, I-frames may be sent along with the transmission of RNR but only the RNR frame should have the poll bit on in this case and it should be sent last in the transmission. The node then listens on the link.

If the receive timeout expires, only the RNR frame with the poll bit on is retransmitted. If an I-frame is heard the RNR frame with poll bit is retransmitted. (Note - the node may accept or ignore the I-frame at its discretion even though it is trying to tell the other side to stop sending them.) If an RR or RNR frame with the P-bit on is heard, the node should respond with I-frames followed by an RNR frame with the P-bit on if it can send I-frames or otherwise by transmitting two RNR frames, one with the poll bit on and the other with the poll bit off and start a receive timeout. (Note that two are required, one to respond to the poll from the other side and one to do a poll for the proper response from the other side. Finally, if an RNR or RR frame with the P-bit off is received this will be accepted by the node as an indication that the other side is aware of the not ready situation and the node can go back to sending I-frames if it has any to send and is permitted to send them. Most of these actions can be deduced from Fig. 4 above.

Note that no node is permitted to send I-frames after receiving RNR and before receiving RR.

Ready condition handling

When a node becomes ready to receive I-frames on a link it will send an RR frame with the poll bit on at the earliest opportunity. If permitted by the other side, I-frames may be sent along with the transmission of RR but only the RR frame should have the p011 bit on in this case. The node then listens on the link.

If the receive timeout expires, only the RR frame with the poll bit on is retransmitted. If an RR or RNR is heard with the poll bit on the node should respond by either transmitting I-frames followed by an RR with the poll bit on or otherwise by transmitting two RR frames, one with the poll bit off and the other with the poll bit on and start a receive timeout. If an RNR or RR frame with the P-bit off or an I-frame is received this will be accepted by the node as an indication that the other side is aware of the ready situation and the node can go back to sending I-frames if it has any to send and is permitted to send them.

FULL-DUPLEX PROCEDURES

A full duplex link requires two independent channels for the transfer of data. The full duplex protocol is designed so that it is possible for both sides to continuously transmit data to the other side simultaneously. One channel is used for transmissions by node A for example and the other channel is used for transmissions by B. Because of the potentially continuous nature of these transmissions, it is usual that each node never listens on its own transmitting channel. However, a node may delay its transmissions if it has some way of knowing that its transmitting channel is unuseable.

This version of V-2 only defines a point-to-point full-duplex protocol. This has uses in 'backbone' linking for an Amateur Radio digital communications network and for satellite work. It is an option which does not have to be implemented in all nodes. Extensions to V-2 for multipoint full duplex operation are being planned but are not within the scope of this paper.

Information transfer

In full-duplex each node listens even when transmitting. Frame numbering and acknowledgment is similar to half-duplex except that acknowledgments are done 'on the fly' even while the other side is transmitting.

A node sends I-frames in sequence until it either has no more frames to send or there are 7 frames outstanding. At this point it will put the P-bit on in the final frame to request acknowledgement from the other side and begin a receive timeout. In order to improve throughput on links with a long turnaround delay - such as satellites, it is permissible to put the poll bit on in a frame before 7 frames are outstanding and still continue to transmit up to the maximum of 7 unacknowledged frames. This technique is called 'pacing'.

If the receive timeout expires meaning nothing was received, the node will not retransmit the previous frames but transmit an RR or RNR frame with the poll bit on and again wait for a response. This operation repeats itself until something is received from the other side. (The higher level of protocol is notified after every N1 consecutive timeouts.)

When a response is received, the node starts transmitting I-frames again, beginning with the first unacknowledged I-frame.

If a node receiving sequenced I-frames encounters a frame not in sequence it should send a

REJ frame at the earliest opportunity it gets. The node receiving the REJ should go back to transmitting the last unacknowledged frame and continue from there. The node should not transmit any more REJ frames until the reject condition is cleared by the receipt of the numbered I-frame indicated in the REJ frame.

Not ready condition handling

Not ready conditions are handled similarly to that described in the half-duplex environment except that the channel doesn't have to be turned around.

Ready condition handling

Ready conditions are handled basically the same as described in the half-duplex procedures.

THE BIG PICTURE

The **datalink** layer of protocol described herein is only one part of a complete communications protocol. As per the suggested IS0 protocol layering structures, this protocol only interfaces with the two adjacent layers in the node in which it is running and with the **datalink** layer on the other side of the link.

The **datalink** layer receives data and orders **from** the Network layer immediately above it. The network layer is a 'higher' level of protocol **and** sort of acts like the **datalink** layer's **'boss'. The datalink** layer acts on these commands and reports on its progress and status as the job is done. When unusual conditions occur which the **datalink** layer **can't** handle it goes to its **'boss'** for advice.

The **datalink** layer passes data and commands to the next lower protocol layer - the Physical layer which in the Amateur environment is usually the HDLC Controller chip. The physical layer acts on the commands from the **datalink** layer and returns data and status information to the **datalink** layer by means of status lines and interrupts.

The **datalink** layer also communicates to the **datalink** layer on the other side of the link.

Even though it is only the third of these interfaces which is the subject of this paper, it should be realized that in a real implementation of a **datalink** protocol there are actually three different protocols involved: i.e. the three communication interfaces. The type of protocols used for adjacent level communication are very system dependent but I will discuss in a general **way,** the type of information that is exchanged between the Network and **Datalink** layers.

Interface with the Network **layer**

Commands to the **datalink** layer and data to be passed **accross** the link are passed from the network layer to the **datalink** layer. The basic commands needed are:

1. Link - This is a command which orders the **datalink** layer to establish a link. The data passed with this command indicates what type of link (either Full or Half Duplex) is desired and the node name of the node to be linked to.

2. Unlink - This is a command which orders the **datalink** layer to terminate a link.

Status information and information received from the link in I-frames are passed to the Network layer from the **datalink** layer. The following status information should be passed:

1. When a **datalink** becomes established. The type of link and node name should be passed as well.

2. When a **datalink** is terminated. The reason code and node name should be passed as well.

3. Whenever a predetermined number of consecutive timeouts **(N1)** on a link has occurred.

4. When an invalid frame is received on a link.

The above lists are not necessarily complete but should serve to give the reader a good idea of the type of information needed. Each implementation will have its own method of handling these interfaces and it is not the intent of this paper to give detailed information as to how the information is to be passed.

SAMPLE EXCHANGE

The following example shows a V-2 exchange with no errors or exceptional situations encountered It is beyond the scope of this paper to show an example of all possible types of **exchanges.** It is hoped that the information in this paper will be sufficient for the reader to determine what the exchanges would be when receive timeouts, lost frames, invalid **frames,** and colliding frames are encountered. The **A,C** and I-fields are shown separated by commas. The address field is displayed in hexadecimal. The control field is represented as follows:

T (Ns) P (Nr)

Where T is the frame type acronym and P is the poll bit. The (Ns) and (Nr) are only shown for frames which contain them and the P is only shown if the poll bit is on **(1).**

The Network header sub-field in the I-field in I frames is shown in Hexadecimal characters and the rest of the I-field is shown in ASCII characters. The subfields in XID and DISC frames are shown separated by commas, the node names being in ASCII characters and the **P,** T and R fields in Hexadecimal. Time progresses downward in the following charts.

The communication is between two nodes. Here is the pertinent information about the two nodes.

Call sign	**VE7APU1**	**KA6M**
Node number	**627B**	68ED

Exchange **1.**

This example shows link establishment, information transfer and link termination. A complete QSO. Arrows to the right indicate transmissions from **VE7APU** and those to the left indicate transmissions from **KA6M.**

68ED627B,XID-P,VE7APU1,KA6M ,P=01,T=00,R=00 ---->

<------ 627B68ED,XID,KA6M ,VE7APU1,P=01,T=00,R=00

68ED627B,I(0)P(0),0100,Hello Hank **---------------->**

<-------------- 627B68ED,I(0)P(1),0100,Goodbye Doug

68ED627B,RR(1) ----------------------------------->

68ED627B,DISC-P,VE7APU1,KA6M ,P=00,T=00,R=00 --->

<----- 627B68ED,DISC,KA6M ,VE7APU1,P=00,T=00,R=00

The first line shows **VE7APU** has entered the link establishment phase at request of the Network level and is trying to initiate (Poll bit is on) a half duplex link (Link Type = 00) with **KA6M** using version 0 level protocol (Protocol = **01**). Because the Poll bit is on we know **VE7APU** has started a line timeout.

The second line shows **KA6M** has entered the link establishment phase and is responding (the Poll bit is off) positively (the Reason field **=** 00) using version 0 level protocol. When this frame is received, **VE7APU** will enter the information transfer phase and cancel the line timeout.

The third line shows **VE7APU** sending an I-frame with data to **KA6M** and demands a response from **KA6M** (the Poll bit is on). **VE7APU** starts a line timeout at this time. When **KA6M** receives this frame it will enter the information transfer phase.

The fourth line shows **KA6M** sending an I-frame with data and acknowledging correct reception of **VE7APU's** I-frame **(Nr=1).** It is demanding a response from **VE7APU** and starts a line timeout.

The fifth line shows **VE7APU** acknowledging **KA6M's** I-frame by sending a Receive-Ready (RR) frame with Nr set to **1.** It has cancelled its receive timeout. The Poll bit is not on so **VE7APU** is not demanding a response from **KA6M** and so does not start a line timeout. **VE7APU** could have sent an I-frame at this point if it had any information to send.

The sixth line shows **VE7APU** has entered the link termination phase on orders from higher levels and is sending a DISC frame to **KA6M** to request termination of the link. The termination has not

been caused by an error condition because the Reason field is 00. VE7APU is demanding a response from KA6M and starts a receive timeout.

In the last line, KA6M enters the link termination phase when it receives the DISC frame and responds (Poll bit is off) with a DISC frame. After transmitting this frame, KA6M goes into the unlinked state.

Since there are no further transmissions on the link we know that VE7APU has received the DISC, cancelled its receive timeout, and gone into the unlinked state.

RESTRICTIONS AND LIMITATIONS

1. This protocol will not function correctly if a node in the information transfer phase can hear frames from another node which is also in the information transfer phase and is using the same four-byte link address. Note that this would require two pairs (4) of nodes active on the same channel each pair using the same area number and node number. One node from each pair would have to be linked with one node from the other pair. Without going into any mathematical analysis I will state that, the probability of this situation arising is extremely low. (Somewhat similar to the probability of the FCC or DOC issueing the same call to two different stations!).

2. Although multiple links are allowed, a link will not be established to two nodes with the same address at the same time. The reason the link is not established is passed to the other node and corrective action could be taken - perhaps by changing the call sign suffix.

3. Multiple links between the same nodes are not allowed. The need for this fieature is questionable as all necessary data traffic accross the link should be able to be multiplexed by the higher levels of protocol. Although not defined in this protocol, a node could masquerade as a different node by supporting multiple node addresses and names by changing the call sign suffix.

4. This protocol only supports datalinks between pairs of nodes. Thus it does not support conferencing or roundtable communication among a group of nodes. This function is normally provided by higher levels of protocol. The complications and problems encountered in providing data integrity with this service at the datalink level are severe and I do not know of any commercial or Amateur datalink protocols which do this. However, a form of roundtable operation could be implemented which does not guarantee data integrity. Each node in the group could operate in unlinked mode transmitting and receiving data in UI frames. The members of the group could agree to all use a broadcast address FFxx (where xx=a group number) in the destination part of the link address and then only pass information from frames which had a link address of this form. Members of the group should be in good communication with each other because there is no way of recovering data from lost frames when using this method of conferencing.

IMPROVEMENTS OVER THE OLD VANCOUVER PROTOCOL.

1. Both full and half-duplex links are provided for.

2. Multiple links are supported.

3. Some multiple protocol support.

4. Vastly increased number of link and node addresses.

5. No coordination of node addresses is required.

6. A rudimentary conferencing system is provided.

COMPARISON OF V-2 AND AX-25

I am including this section because I am sure that many people will make comparisons and also because AX-25 is the only other Amateur Radio packet protocol which has a published specification document.

This comparison is difficult to do because despite statements in the literature that AX-25 is a link level protocol, it is, in fact, a type of network level protocol. AX-25 routes packets through a series of links from a source node to a destination node. Although AX-25 provides end-to-end flow control, sequencing and data integrity, it provides no link level error recovery, retransmissions, acknowledgements, sequence checking, etc. for any of the links in the chain. Only in the special case where the source and destination are connected by a single link could AX-25 be construed to be a link level protocol. In this special case, the end points of the connection become identical to the end points of the link and so the end-to-end connection facilities cannot be distinguished from the link facilities. Some comparison of the two protocols can be done when considering this special case.

A document describing a Network level protocol for V-2 will be published later. Routing of packets through a network is one of the responsibilities of the Network level. Code for a Network level is being written at the present time for the VADCG TNC. Code for this link level protocol has already been implemented on the VADCG TNC.

1. V-2 has a reduced protocol overhead compared to AX-25. A 4-byte fixed length address field as opposed to a 14-byte or greater address field in AX-25.

2. V-2 has a facility to select full-duplex or half duplex protocols at link establishment.

3. V-2 has a facility to select different protocols at link establishment.

4. V-2 uses an addesssing and naming system for nodes. AX-25 makes no distinction between names and addresses.

5. AX-25 has a network routing system. V-2 link level does not.

6. V-2 does not bit shift call signs and other names when transmitted.

7. V-2 makes use of the selective receive function of the SDLC/HDLC protocol controller chips to automatically eliminate frames that do not need to be checked. AX-25 cannot do this. For example, in this area, all call signs start with the character 'V'. Using AX-25, this would mean that all frames transmitted in this area would have the same character after the initial flag in every frame. This effectively renders useless the selective receive function present on the protocol controller chips and forces the software to read into memory and examine every frame that is transmitted on the channel.

SUMMARY

The general specifications of the V-2 protocol have been presented in this paper. As this is the first draft of a new protocol, some details may have been omitted. This document will be revised as necessary, to expand on areas not clearly presented and to include changes in the protocol. Those interested in obtaining the latest version of this document should contact the author. Anyone having questions or comments should do the same, preferably by writing.

At the present time most of this protocol has been implemented in the VADCG TNC. The source code for this implementation is available on CP/M 8-inch diskettes. Anyone wishing to implement it on another system should contact the author directly so that the work may be coordinated.

The author feels that V-2 link level is an efficient protocol both in terms of channel utilization and software requirements and that it is particularly suited to the Amateur Radio environment. It provides almost all the functions required by the IS0 data link model without overstepping into functions in the domain of other layers. It is intended as a building block upon which higher levels of protocol can be independently developed as per the IS0 proposals.

REFERENCES

1. Rutledge, J. H. "OSI and SNA: A Perspective" IBM Washington Systems Center Technical Bulletin, April 1981, GG22-9225-00

2. Data Processing - Open Systems Interconnection - Basic Reference Model, International Standards Organization Draft Proposal 7498, December 1980

3. Fox, Terry "AX-25 Level 2 Protocol" Second ARRL Amateur Radio Computer Networking Conference proceedings, March 1983

4. Fox, Terry "Alterations To AX.25 Level 2 Document By ARRL Digital Comm." AMRAD newsletter, November-December 1983

5. "Using the 8273 SDLC/HDLC Protocol Controller" Intel Peripheral Design Handbook, August 1981

APPENDIX A. Node Address Calculation

The CRC-16 algorithm divides the 7-byte character string of the Node Name by the following generating polynomial:

$$x^{16} + x^{12} + x^5 + 1$$

This is the same polynomial used in calculating the FCS field by the various HDLC/SDLC protocol controller chips. In the calculation, integer quotient digits are ignored and the 16-bit remainder is checked to see that the first byte is not FF (Hexadecimal) and if it is, the bytes are reversed. If the first digit is still FF the value used is 0000. This playing around with the value is done to ensure that none of the special purpose addresses beginning with FF are generated.

Many of you out there may find this a little difficult to understand so I am including a sample assembly language program listing below which actually does the above generation for an 8080 microprocessor. It should not be difficult to implement in other processors.

Program listing

```
                    ; ROUTINE TO GENERATE A NODE ADDRESS FROM A NODE NAME
                    ; WHEN CALLED, IT USES THE NODE NAME AT LOCATION 'NODENM' AND
                    ; LEAVES ITS NODE ADDRESS AT 'CRC' UPON RETURN
                    ; IT DOES NOT GENERATE ADDRESSES WITH FF AS THE FIRST BYTE

0000                ADRCALC:ORG     0
0000 210000                 LXI     H,0       ; INITIALIZE CRC FIELD TO 0000
0003 225100                 SHLD    CRC
0006 215300                 LXI     H,NODENM        ; POINT TO 7-BYTE NODE NAME
0009 1607                   MVI     D,7       ; NUMBER OF CHARACTERS
000B 7E             LOOP:   MOV     A,M       ; GET A CHARACTER
000C CD2D00                 CALL    CALCCRC   ; GIVE IT TO CRC ROUTINE TO PROCESS
000F 23                     INX     H         ; POINT TO NEXT CHARACTER
0010 15                     DCR     D         ; HAVE FINISHED WITH ALL THE CHARACTERS?
0011 C20B00                 JNZ     LOOP      ; NO, GO BACK TO PROCESS ANOTHER
0014 2A5100                 LHLD    CRC       ; GET THE GENERATED CRC ROUTINE
0017 7D                     MOV     A,L       ; IS THE FIRST BYTE FF (HEXADECIMAL)?
0018 FEFF                   CPI     0FFH
001A C0                     RNZ               ; NO, RETURN WITH NODE ADDRESS IN CRC
001B 7C                     MOV     A,H       ; YES, BAD LUCK, REVERSE THE CRC BYTES
001C 6F                     MOV     L,A
001D 26FF                   MVI     H,0FFH
001F 225100                 SHLD    CRC
0022 7D                     MOV     A,L       ; NOW IS THE FIRST BYTE FF?
0023 FEFF                   CPI     0FFH
0025 C0                     RNZ               ; NO, RETURN WITH NODE ADDRESS IN CRC
0026 210000                 LXI     H,0       ; YES, EXCEPTIONALLY BAD LUCK
0029 225100                 SHLD    CRC       ; SET NODE ADDRESS TO 0000
002C C9                     RET               ; AND RETURN, JOB COMPLETE.

                    ; THIS IS THE ACTUAL CRC-16 CALCULATION ROUTINE
002D E5             CALCCRC: PUSH   H
002E 0608                   MVI     B,8
0030 4F                     MOV     C,A
0031 2A5100                 LHLD    CRC
0034 79             CALCCRC1: MOV   A,C
0035 07                     RLC
0036 4F                     MOV     C,A
0037 7D                     MOV     A,L
0038 17                     RAL
0039 6F                     MOV     L,A
003A 7C                     MOV     A,H
003B 17                     RAL
003C 67                     MOV     H,A
003D D24800                 JNC     CALCCRC2
0040 7c                     MOV     A,H
0041 EE10                   XRI     10H
0043 67                     MOV     H,A
0044 7D                     MOV     A,L
0045 EE21                   XRI     21H
0047 6F                     MOV     L,A
0048 05             CALCCRC2: DCR   B
0049 C23400                 JNZ     CALCCRC1
004C $5100                  SHLD    CRC
004F                        POP     H
0050 C9                     RET

0051 0000           CRC     DW      0    ; WHERE NODE ADDRESS IS PLACED
0053 5645374150NODENM DB    'VE7APU1'       ; NODE NAME

005A                        END
```

WORKING "PACKET" ON OSCAR 10

H. S. Magnuski, KA6M
311 Stanford Avenue
Menlo Park, CA 94025
415-854-1927

Abstract

This paper summarizes the technical and operational aspects of trying to communicate via packet radio on the AMICON channel of AMSAT's Oscar 10 satellite. A calculation of effective throughput is made which considers factors other than the traditional Eb/No parameter.

Background

In the summer of 1983 the Phase IIIB satellite of the Radio Amateur Satellite Corporation (AMSAT) was launched and became known as Oscar 10. After some early problems, the satellite achieved a stable, highly elliptical orbit of 26 degrees inclination. The orbit achieved provides outstanding communications coverage, and multi-hour path openings between points on the globe.

Within hours of the turn on of the transponder, several well equipped amateurs who were both packet radio & satellite enthusiasts attempted to transmit digital information on the special channel reserved for packet radio use. The opening of the transponder was also a challenge for the rest of us (primarily computer types) to take advantage of this new communications resource and to learn what the words "signal-to-noise ratio" really mean.

The author started assembling a satellite station in earnest in the Fall of 1983, and even though it would make a good story, this paper will concentrate on experiences of using the satellite, which has been the main activity since last December, when the last piece of the antenna system was put into place.

The plan is to describe as many of the technical and operational parameters involved in using the satellite for digital communications as is possible. These comments and observations are in the form of a calculation which attempts to compute the probability of getting a packet through the satellite at any given instant. Please don't take the figures too seriously, as most of the numbers are approximations and there are obvious holes in the logic used to compute probabilities. Nevertheless, the reader should gain an appreciation of the technical challenges involved in working packet on Oscar 10, and see how much room there is yet for innovation and improving our systems.

The sections which follow discuss the various factors involved in working packet, and in each case a number is derived which states the probability that the factor does not interfere with the successful transmission of that packet. At the end of the paper all the probabilities are multiplied (independence of factors is assumed) to determine the overall probability of success.

Satellite Visibility

The satellite is in a highly elliptical orbit, with an apogee of 35,000 km. and a perigee of 3500 km. The apogee sub-satellite point moves east about 9 degrees in longitude per day, and the apogee sub-satellite latitude remains constant at around 25 degrees. There are 2.06 orbits per day, but usually only one is visible for a given site, except at the transition where eastern passes change over into western passes, Antenna position during apogee is relatively stable, and only fine tuning is required. A printout or computer program is required to find out exactly where the satellite is during a pass, but the pattern is quite regular, and after a while one can almost sense where to point the antennas for the next orbit.

A complete pass lasts almost ten hours, but during perigee the transponder is turned off, and experience has shown that packet work is only good around apogee plus or minus 2 1/2 hours or so. So the channel is only usable for about 5 out of 24 hours, leading to a visibility probability of 5/24 or 0.21 .

Normally the satellite is in Mode B (435 MHz. up, 145 MHz. down), but for four hours around apogee on Wednesdays and Saturdays it is put into Mode L (1269 MHz. up, 436 MHz. down) which is currently unusable for packet work. So the Mode L time is 8 hours in a 168 hour week, and thus the Mode B probability is 160/168 or 0.95 .

A complete western horizon to eastern horizon series of orbits takes about twenty days, and the antennas point into a hill on the low western passes, and into an oak tree on the low eastern ones. So scratch about four days out of every twenty, and the probability of not irradiating the neighbors becomes 16/20 or 0.8 .

Every month the orbits for the next month are computed at work and printed for 30 days in advance, but usually only after getting ready to work the satellite and finding that the calendar has expired. So one day out of thirty is lost, and the probability of having the right printout is 29/30ths or 0.97 .

As the orbit apogee passes east over the 180 degree azimuth mark and works its way into Europe, the big boomers over there desense the receiver and make the power levels required to work packet more than the average ham can afford. So the probability of no Europeans is about fifty percent or 0.50 .

The earthstation ground terminal equipment room is also the OM's and XYL's bedroom, and this means that late night passes are unusable. So, the probability of daylight is 16/24 or 0.67 .

• = ◦ finances required to buy all this satellite equipment required the author to maintain his job, and the remaining time is roughly split between playing with the kids and playing with the radios . So, the probability of having some free time and not arranging a PPRS meeting or talking to a local ham club is perhaps 4/16 or 0.25.

Interference Conditions

One of the major disruptive elements to the success of working with the satellite today is man-made electrical noise. This is particularly troublesome on packet, where a stray impulse can kill a complete frame. The main sources are autos, line noise, and the neighbors' mixmasters, electric vibrators and buzzsaws . Two meters is very prone to interference of this type, and a move to higher frequencies would be warranted for this purpose alone. The subjective impression is that roughly twenty percent of the time some static is in the air, and thus the no-noise probability is 0.80 .

The packet downlink is 145.830,which happens to be the favorite watering hole of some crusty old simplex fm'ers in this area, who can't imagine why anybody would want to use their frequency, Most will move when asked, but others can't seem to comprehend that a five-way international qso could be in progress when the local channel sounds perfectly clear. The crusties seem to get their way about ten percent of the time, so the no crusty probability is 0.90 .

The receiving equipment has to be extremely sensitive, and unfortunately will pick up lots of stray radiation from unfiltered computer gear and unshielded, unboxed terminal node controllers, All the equipment purchased should be FCC Class B Certified, and all connecting cables should be shielded and grounds well maintained. One birdie from my modem on 145.760 serves as a nice test signal source, but the rest may cause trouble five percent of the time. Therefore, the non-birdie probability is 0.95.

Station Equipment

In order to get anywhere on packet, a decent antenna system is required with a mast-mounted pre-amp. The commercially available gas-fets have • S dB noise figures, and are not too expensive, Most hams do not bother putting coax switches in their two-meter downlinks and setup their equipment so that the probability of transmitting into the gas-fet is near zero. Here are three creative excuses to tell friends after you've just smoked your high-priced gas-fet pre-amp: "My TAPR TNC did a cwid when I didn't expect it", "My two-year old was playing with the microphone", and the classic "My 726 was on RB-TA instead of RA-TB". So the probability of having a functioning gas-fet is about 0.98 .

Most packet stations are currently using the AX.25 protocol at 1200 bps with 202-style FSK modems. There are a few stations who may still be using older protocols or different modems or slower speeds. The compatibility probability is therefore 0.95 .

The use of 202 FSK modems is currently the biggest barrier to successful packet work, as their theoretical and practical performance is poor in relation to other potential designs, Tuning is difficult, and different designs of the 202 have different performance levels. If one has stable equipment, a tuning indicator such as an "eye" pattern, and reasonable receiving equipment , then about nine out of ten packets can be recorded successfully. So, the frame throughput probability is 0.90 .

Channel Contention

The packet community is relatively small at the time of this writing, only about a dozen or so stations have actually transmitted so far, So the probability of collisions with other traffic on the channel is quite small at the present time. However, the quarter second round trip delay permits two transmitters to start and collide without hearing each other's carrier, and this occasionally happens, Another source of collisions is TAPR boards left in full-duplex mode while running on a single frequency (not-split frequency) channel. So an estimate for the probability of no collisions is 0.95 .

Net Throughput

If the above factors are considered independent, then the probability of a successful transmission is the product of each of the above factors. Multiplying all the items above, we compute a net transmission probability of • 007059e This may not seem very high, but considering that twelve months ago the probability was zero, it is an infinite improvement!

If we also consider that in just one week there are 604800 seconds of potential transmission time on which we can place 72,576,000 bits of information, or 2,268,000 forty byte frames, then using the net number above implies that 16,009 forty byte frames will get through the channel in a week. That's 640,392 characters of information, more than an average person can consume in a week given reasonable assumptions,

Conclusion

Using the satellite for digital work has been one of the most interesting technical challenges the author has encountered for some time. It makes one appreciate the role satellites will play in the future, and opens up many new and interesting possibilities for information transfer,

Clearly, new designs in modems and automation of the ground station control systems will vastly improve the throughput available on this resource, These are interesting technical challenges which lie ahead, and others are encouraged to add their contributions.

A PACKET RADIO EMERGENCY COMMUNICATIONS SYSTEM

Bob Neben K9BL
126 E. Schantz Ave.
Dayton, Ohio 45409
(513) 299-4436

Introduction

We have come a long way in the use of Packet Radio. **In** the past few years we have gone from a handful of experimenters **proving** the practicality of the concept, to hundreds and soon thousands of active Packeteers. Talking to one another to help the synergism of ideas is valuable, but the time is now to start building a viable system that will help the public good.

Topologies

We have an assortment of ways to communicate in Amateur Radio. The thing we do best is talk to one another i.e. one ham talking with one other ham. The media may be 2 meter FM, HF SSB, RTTY, SSTV, or whatever, This is the Point to Point topology (Figure 1).

```
**********ox*            *************
*           *            *           *
* Station 1 * ---------- * Station 2 *
*           *            *           *
*************            *************
```

Figure 1.

This is the best communications network ever devised. There **is** only one conversation on the frequency; when one station is talking the other station is listening. There is no interference but should a retransmission be required, the communication could easily be handled by either station. The chances of the data being sent incorrectly is low because either can ask for **retransmission, clarification** or **additional information** if necessary. This is the typical amateur transmissi one However, we also use other topologies.

```
*******          *******          *******
*  1  * -------- *  2  * -------- *  3  *
*******          *******          *******
   !                                 !
   !                                 !
*******          *******          *******
*  6  * -------- *  5  * -------- *  4  *
*******          *******          *******
```

Figure 2.

The roundtable is a ragchewing mode used with a group of operators. Each person keeps a 1 ist of stations so they know which one to pass it to next. Other stations can break in to the conversation if desired or **a** station could drop out, however, it is courteous to sign off. Conversations tend to reflect on what the last person said, since individual stations do not normally keep notes, This is the Ring topography (Figure 2) and it has limited usefulness since no station is "running the show".

```
                 *******
                 *  2  *
                 *******
                    !
*******          *******          *******
*  5  * -- * 1  * -- *  3  *
*******          *******          *******
                    !
                 *******
                 *  4  *
                 *******
```

Figure 3.

The directed net is typical of traffic handling situations. This is the Star topology (Figure 3). The net control station directs all traffic and no communication takes place without prior approval. This is a good way of getting the traffic to **its** destination, but at a high cost in terms of human efficiency,, Typical nets have upwards of a dozen or more stations. Since only two stations can converse at the same time, the remainder must just listen to a lot of traffic that does not apply to them, except of course for announcement5 or bulletins. The efficiency of the net is terrible, As the number of stations increase and the traffic volume increases, the efficiency drops still further. During slack periods or when the volume of traffic eventually diminishes, these many operators ask themselves (and rightfully so) "Am I really needed here?" Depending on net discipline and managerial techniques, a net could 1 ose many operators just before they are most needed. Worse yet, the operators may stay around for that weather watch but won't show up for the next one. There must be a better way.

I want to mention another kind of net: one that doesn't exist in amateur radio - yet. It's not implemented yet because it only applies to digital networks, Any station can transmit on the frequency (bus) and all stations are listening at all times. Because of the microprocessor present, stations will only listen to what the station wants to listen to or is directed to Listen to by the human operator. Station 1 can converse with station 5. Meanwhile station 2 can talk to stations 3 and 6. Station 4 can monitor everyone and no station need listen or even be aware of any other station's conversation. This is the Bus topology (Figure 4), but as far as the individual stations are concerned, it looks like the Point to F'oint topology (Figure 1). This is packet radio at its best and I would like to show you how you can apply this to emergency communications.

```
*******     *******     *******
*  1  *     *  2  *     *  3  *
*******     *******     *******
   :           :           :
   ======================================
         :           :           :
      *******     *******     *******
      *  4  *     *  5  *     *  6  *
      *******     *******     *******
```

Figure 4.

Disaster Operations

There is no such thing as a typical disaster as various officials will confirm; each one is different and unique. However, we can take a typical situation such as a flood. A flood affects a large area to some degree, but the flood is disasterous to only a localized area at any one time. This area is often densely populated although geologically limited to a few square miles. Consequently it affects many people, The first priority is warning these people of danger and if necessary, evacuating them to shelters. Then comes monitoring conditions, maintaining the shelters, and finally cleanup. When things are habitable again the people return to their homes and the shelters close down. The emergency is over.

The type of radio activity varies widely during the operation. Standard operating procedures include lots of overkill and inefficiency but the job gets done somehow. Let's analyze the situation to see if there's a better way.

As soon as conditions warrant the Emergency Coordinator (EC) or their designee goes into the area and establishes the net in the temporary or permanent Emergency Operators Center (EOC). Local officials should already be colocated and have communications of their own to local public services including Red Cross and other agencies. Although slower, telephone service to these agencies can keep the amount of radio traffic manageable. Often however, telephone service is either very limited or unavailable.

The EC communicates over 2 meter FM to various individuals or teams that are assigned to public officials, Red Cross, Damage Assessment Volunteers or Shelter Managers. Sometimes our 2 meter communications is more reliable than that used by these various agencies. The problem comes from density of traffic,

The EOCs tend to be beehives of activity. Everyone wants to head the effort to get the job done. Your group will be getting communication requests from all these agencies for everything from trivial to critical. It's near impossible to say "no" to the mayor. It is our experience that the group with a good handle on this type of activity is the Red Cross. They have the disaster plans and experience and they work very well with groups such as ours. They also act as a clearinghouse on health and welfare traffic.

The communication volume of traffic within the disaster area is higher than anywhere else. The farther you get from the disaster area, the less volume of traffic. With voice communication, there is no choice but to impact this high volume of traffic in the EOC. The high volume of traffic continues in the EOC and surrounding area, however, people outside the area also get on the same repeater or frequency and make the rest of the net wait. Remember, only so many stations can actively be on a net with their traffic at one time before the frequency becomes saturated. The outlying stations with priority traffic are just as important as EOC priority traffic. Getting the activity away from the EOC doesn't help unless you can get that traffic off frequency also. We were partially successful by using 220 MHz as an "administrative frequency", but that meant listening to voice conversations on two radios. Another 2 meter frequency rig won't help because it will overpower the main 2 meter frequency and block reception. Is there a solution to this dilemma? Yes, packet radio! But how do you implement it?

Voice and Packet

For fast communication, there will never be a replacement for voice. Do not even think about asking the mayor to please type his or her message, So the net at the EOC will continue on 2 meters utilizing a base station and operators with mobile or hand held radios doing their thing. What we can do to relieve the bottleneck at the EOC is to establish an effective colocated packet radio system. How do we do this and how can it be used effectively? Remember the bus!

Lots of traffic can be digitized including all routine requests, shelter traffic, Red Cross inquiries, damage assessment reports, ARES status, etc. We must be able to send our packets without affecting the 2 meter voice traffic or disturbing the EOC operators. I propose a parallel system running the traditional voice on 2 meters FM and packet on 220 MHz (Figure 5).

```
******************  **********  **********
* Local  * ARES   *  * Red    *  * Shelter*
* EOC    * EOC    *  * Cross  *  * K9BL-2,*
*        * K9BL-0 *  * K9BL-1 *  * 3, & 4 *
******************  **********  **********
       |                 |            |
     ====================================
     |            |             |
***********  ***********  *************
* Library *  * Gateway *  * Damage    *
* Computer*  * HF Link *  * Assessment*
* K9BL-10 *  * K9BL-11 *  * K9BL-5    *
***********  ***********  *************
```

Figure 5.

I chose 220 MHz for several reasons, 220MHz can transmit without interfering with 2 meter reception and visa versa. Most scanners cannot receive 220 MHz so confidentiality of the information is somewhat protected. Also, it is impractical to simulcast the packet over the repeater voice - the packet attracts too much curiousity and it tends to splatter on to adajacent channels.

The 2 meter voice net would be handled pretty much business as usual, with a few exceptions. Routine requests should be significantly reduced and there will be fewer people out there doing a better job. It may be hard to convince disaster planning councils that they can get more service with fewer people.

EOCs usually have at least two people operating radios. One person serves as net control. and the other interfaces with officials;? monitors conditions? maintains status boards, etc. It is usually difficult if not impossible for one person to serve all these functions. What's needed is one operator to be net control while the other operates the keyboard. Ideally, the keyboard operator screens the requests so only the urgent information ties up the repeater.

Lots of information can be transferred via packet and a record of the traffic can be recorded to disk at one of the stations. If an item demands immediate attention at a particular station, the sender can ring the bell on that persons keyboard. Most traffic, however, will fall in the categories of either inquiry, status or update. One of the major differences between the voice net and the packet net is the lack of net control on the packet net. The packet net operates on the bus topology. However, every station should use the call sign designated for use during the emergency.

Any station can initiate an inquiry. Usually an inquiry is directed at the likely respondent, but perhaps it should go to everyone. If every station uses their own call, we do nat have a vehicle far an all-call. If they use a particular call sign for the duration of the emergency, such as the club call or repeaters trustee's call, then the extensions 0-15 take on a new meaning. We can call selectively (i.e. K9BL-3) or all call (K9BL). Status and update requests, however, implying interfacing a computer.

Computers and packet radio go hand in glove. By using a database program on our home or club microcomputer, we can manage disaster information like it has never been done before. Gone forever are the little scraps of paper all over the EOC. Instead we have neat, organized files that can be called out immediately by any station. It's a lot more professional to check a listing rather than searching through a yellow pad. Chances are a computer listing will be more accurate and up to date, too. All these neat things using computers are of no avail if we don't have a standard message format.

Message Traffic

How do we handle messages? We really don't want to put every message into an editor to rearrange it for our database. What is needed is a standardized format, a standardized database program, and a program to check the incoming messages for compatibility. If necessary? a human could rearrange the message and/or ask for missing information. It would be nice to keep the manual intervention under 10%.

Standardizing a format is more difficult than it looks. Remember we will want to have gateway access into this system. The ARRL messagegram does not adapt very well to packets. No longer do we worry about wordcount since we have our Frame Check Sequence guaranteeing error free reception. The same goes with sequence number since the computer can add the date/time. Calls are even handled automatically. But heading? text, and ending information needs to be standardized,

The military message format is left over from the teletype and adapts nicely to computers. The message contains heading information that could be added by the computer including handling instructions, originating station, date/time group? precedence (default to routine), and destination/addresses. The entire text is free form and the message ends with an ending sign. This is ideal for computers!

When the computer sees a message coming, it assigns it ta a file based on the header information which include:; type of message, date/time, originating

station, precedence and addresses. The text portion depends on the database the message will be in; i.e., damage assessment, shelter listing, etc. The error checking program needs to flag any discrepancies in this.

Conclusion

These messages and associated programs will form the data base that can be examined by any of the packet stations desiring information. Within a short period of time these data bases will contain a large amount of accurate information that will greatly aid the disaster effort and keep the workload manageable on the voice net. We will then be attaining a degree of efficiency never before realized, while serving the needs of our community.

AN APPLICATION NOTE DESCRIBING A LOW POWER RS232 LIKE INTERFACE

Paul Newland, ad7i
Post Office Box 205
Holmdel, New Jersey 07733

Introduction

Radio Amateurs are beginning to make use of low-power (LP) micro-processor systems for controllers and now have need for a LP serial interface to connect them to other LP terminals or computers. This application note describes a LP serial interface that is compatible with conventional RS232 terminals plus the new "lap" computers that have only a "sort-of" RS232 serial interface.

Problem Statement

The "loose" global functional requirements of RS232 are that a MARK or INACTIVE signal will be transmitted as a voltage more negative than -5 volts from a low impedance source and a SPACE or ACTIVE signal will be transmitted as a voltage more positive than +5 volts from a low impedance source. A received signal into an impedance of more than 1 Kohm that is more negative than -3 volts will be considered MARKING or INACTIVE and a signal more positive than +3 volts will be considered SPACING or ACTIVE.

Most implementations of a RS232 interface for amateur radio have used the 1488 driver and 7489 receiver. Both these devices provide an interface compatible with RS232 specifications but neither part can be considered "low-power." An alternate interface implementation is required for a LP system.

An additional problem to overcome is that posed by some of the new lap computers, the Radio Shack Model 100 being a notable example. Some of these lap computers are using non-inverting, 0-3 volt buffers, for their "RS232" interface drivers and as a result, they will properly receive signals from RS232 compatible equipment but may not properly drive RS232 compatible equipment.

The goal of this application note is to provide an interface that will function well with both these standard and sub-standard interfaces.

Implementation

A simple and low cost driver and receiver can be formed with LM324 operational amplifiers. The power supply current drive for one package of four op-amps is less than 1 mA under no-load conditions. Two packages are required, one for receivers and one for drivers. This can be reduced to one package if some output voltage limiting is provided for the receivers. For the receivers, the power for the op-amp should be taken from a +5 and ground voltage source. Power for the drivers can be most anything as long as it is bipolar. The positive supply of the driver op-amps should be taken from a +5 to +15 volt supply. The negative supply of the op-amp can be taken from a -5 to -15 volt source. If the micro-processor system doesn't use a negative voltage supply one can be developed using the Siliconix 7661 configured as a voltage inverter. The voltage range of the driver's output is from the positive supply, less 1.5 volts, to the minus supply (i.e., for +/- 5 volt supply the output range of the driver would be +3.5 to -5 volts).

A reference signal is developed for a slicing/limiting point for use by the drivers and receivers. Figure 1 shows how the reference is generated by R1, DI, D2, and an op-amp. The output of this op-amp will rest at about 1.4 volts above ground with a low impedance.

Figure 2 shows how the output drivers are configured. They function simply as an inverter and bipolar driver. The output will swing from the positive supply (less 1.5 volts) to the negative supply. R1 provides current limiting.

Figure 3 shows how the input receivers are configured. Note that receivers, unlike the drivers, have their power leads connected to +5 volts and ground. The slicing/limiting point is the reference derived in Figure 1. If this point was ground, instead of 1.4 volts, the interface may not work with those terminals using o-3 volt output drivers. Diodes D1 and D2 provide voltage limiting causing the inverting input to be constrained to voltages between 0.7 and 2.1 volts. R1 provides the input impedance control and current limiting while R2 provides default signal conditioning for leads that are often unterminated (i.e., DSR, DTR, RTS, CTS, etc.). R3 and R4 provides hysteresis while R5 pulls up the receiver's output to +5 volts to ensure compatibility with CMOS circuitry.

Conclusion

Using a serial RS232 interface such as this, the idling current should be less than 3 mA, depending on load. The interface is simple to construct and IS compatible with RS232 drivers/receivers as well as TTL/CMOS level (0-3 volts) drivers often found in the new low-power "lap" computers. Additionally, default conditions can be set for unterminated receiver inputs.

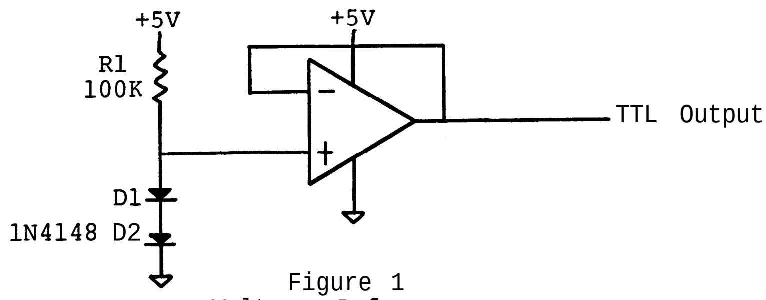


Figure 1
Voltage Reference

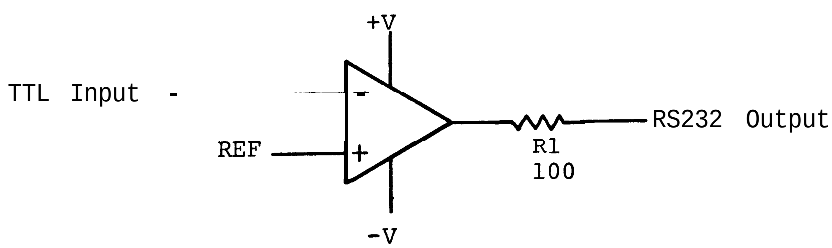


Figure 2
Driver

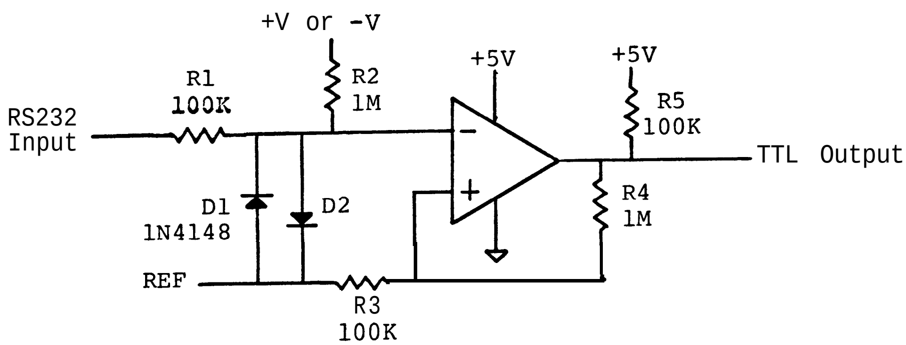


Figure 3
Receiver

Q-CALL
A Method of Providing Selective Calling for AMTOR Using Mode-B Collective Broadcast (Bc)

Paul Newfand, ad7i
Post Office Box 205
Holmdel, New Jersey 07733

1. Introduction

This document defines a method of providing a Selective Group Calling (SGC) facility on top of CCIR Rec 476-3 (AMTOR) MODE-Bc[1]. SGC has application on 80 meters and VHF where propagation provides consistent communications. It provides a mechanism to allow a group of stations to intercommunicate without printing messages of other groups or individuals sharing the channel,

There are two major features of a SGC facility for CCIR 476 that must be provided regardless of the final facility definition. First, the facility must be, or have the potential to become, a recognized amateur radio standard. Secondly, it must be possible to provide the facility either within a CCIR 476 code converter or external to it. Below is a proposal that I feel meets these needs. First the major facility requirements and constraints are outlined then the proposed transmission and reception procedures are given.

2. Requirements

The following items were considered to be requirements for the facility. Additional requirements may be added at a later time but those listed below should be considered fundamental.

2.1 Labeling

The method used to implement SGC should have a distinctive label in the RTTY terminology. I have chosen Q-CALL as a label because it is a literal description of the proposed calling mechanism and it is unique to RTTY terminology. Q-CALL uses for its SELCALL the letter Q followed by the 4 non-control characters of the CCIR 476 "call" signal.

2.2 Resistance to Falsing

The sequence of characters that forms the SELCALL for SGC should be something that would be unlikely to occur during normal communications.

2.3 Group Calling

The facility should have the ability to concatenate SELCALLs so that a true "group" calling capability exists rather than simply the ability to send one SELCALL that many stations respond to. When one SELCALL is shared among several users the calling station can not change the groupings; this is undesirable.

2.4 Generation of SELCALL Information

The SGC SELCALL string should be such that it can be generated either internal or external to the CCIR 476 code converter without critical timing requirements.

2.5 Detection of SELCALL Information

The SGC SELCALL string should be such that it can be detected either internal or external to the CCIR 476 code converter without critical timing requirements.

2.6 Generation of End-of-Communications Signal

The SGC *'end-of-communication*' signal should be such that it can be generated either internal or external to the CCIR 476 code converter.

2.7 Detection of End-of-Communications Signal

The SGC *'end-of--communication" signal should be such that it can be detected either internal or external to the CCIR 476 code converter.

3. Q-CALL Proposal

The signaling to provide these features can be simple to implement. What follows are the current recommended procedures for the Q-CALL SGC facility.

3.1 Generation of SELCALL Signals

Each Q-CALL SELCALL will be formed by a CR, LF, LTRS, and the following sequence repeated 8 times: "Q" "wxyz" "SPACE" . Within the sequence "SPACE" is defined to be the character SPACE, "Q" is defined to be the character Q and "wxyz" is defined to be the four non-control characters of the CCIR 476 "call" signal normally used for MODE-A communications. For group broadcast, several SELCALLs may be concatenated by sending the first, then the second[2], and so on until all desired SELCALLs have been transmitted. During "SELCALL GENERATION" the inter-character time may not exceed 30 seconds.

3.2 Generated Message Content

All messages must begin with a CR or a LF character. The message may be in any format except that the sequence NNNN is prohibited and during the message transmission the inter-

character time may not exceed 30 seconds.

3.3 Generation of End-of-Communications Signal

End-of-communication will be signaled by the transmission of the character sequence NNNN followed by the CCIR 476 MODE-Bc end-of-communication signal (3 alpha characters).

3.4 Detection of SELCALL Signals

When the Q-CALL processor detects the sequence: "Q" "wxyz" "SPACE" "Q" "wxyz" "SPACE" "Q" "wxyz", it will set a SELCALL-DETECT flag[3] indicating that a valid Q-CALL SELCALL has been detected, provided that the inter-character time of the received characters does not exceed 45 seconds. With in the Q-CALL sequence "SPACE" is defined to be the character SPACE, "Q" is defined to be the character Q and "wxyz" is the four non-control characters of the CCIR 476 "call" signal normally used for MODE-A communications by the receiving station. When the Q-CALL processor detects that a CR or LF character has been received and that the SELCALL-DETECT flag is set, a DATA-OUTPUT flag will be set.

3.5 Message Reception

Each teleprinter character received by the Q-CALL processor will be passed to the teleprinter[4] while the DATA-OUTPUT flag is set.

3.6 Detection of End-of-Message Signal

When no characters have been received for 45 seconds, or the character sequence NNNN has been received, or the CCIR 476 MODE-Bc end-of-communications signal (3 alpha characters) has been received the DATA-OUTPUT and SELCALL-DETECT flags will be reset.

4. Rational of Q-CALL Mechanism for Transmission and Reception

4.1 Selection of SELCALL Signals

Any SGC SELCALL built on top of AMTOR should be something that makes use of the MODE-A "call" signal and is easily readable by an operator. It should be all letters case so that the Q-CALL processor does not need to worry about handling FIGS and LTRS characters[5]. The call should be repeated, without LTRS, FIGS, or any other extraneous characters for at least 8 repetitions. Eight repetitions ensures that the SELCALL will have a high probability of being detected during less than optimum propagation conditions. CCIR 476 "call" signals are often sent using MODE-Bc and repeated several times with only one space separating them[6]. For that reason, I have included a "Q" and "SPACE" as part of the SELCALL string for a visual delimiter rather than just a space. Including a CR and LF at the beginning of each SELCALL ensures that detectors provided external to a CCIR 476 code converter will "see" the SELCALL[7]. It also ensures that anyone monitoring the channel with conventional MODE-Bc equipment will be capable of reading all SELCALLs that are part of a group without having the printer over-write information at the end of a line.

4.2 Selection of End-of-Communications Signal

The NNNN is easily detected by external equipment and the standby condition is easily detected internal to the CCIR 476 converter. The time-out feature is added to aid stations using an external Q-CALL detector.

5. Procedures for a Q-CALL Transmission

The Q-CALL transmission sequence should be functionally as follows:

1. Set code converter to MODE-Bc transmit.
2. Send CR, LF and LTRS.
3. Send 8 repetitions of the "selective call" (ex., ... QAADI QAADI QAADI ...).
4. If necessary send another new-line sequence and send any other SELCALL sequences needed. Repeat this step until all SELCALLs have been transmitted.
5. Send the message (the first character of any message must be either a line-feed or carriage-return).
6. Send NNNN and switch to standby.

EXPLICIT EXAMPLE

```
=============================================
((transmitter on, MODE-Bc))

QRA AD7I
QAABC QAABC QAABC QAABC QAABC QAABC QAABC QAABC
QWXYZ QWXYZ QWXYZ QWXYZ QWXYZ QWXYZ QWXYZ QWXYZ
QWABC QWABC QWABC QWABC QWABC QWABC QWABC QWABC
QWDEF QWDEF QWDEF QWDEF QWDEF QWDEF QWDEF QWDEF

AB12C, W2XYZ, W2ABC, W2DEF DE AD7I

A NEW STATION IS NOW ACTIVE ON THIS FREQUENCY.
W2GHI IS ON USING SELCALL WGHI FROM NEW YORK
CITY.  W2GHI IS OPEN MODE-A FROM 7PM TO 9PM
LOCAL TIME AND OPEN FOR Q-CALL 24 HOURS A DAY.

AB2C, W2XYZ, W2ABC, W2DEF DE AD7I
NNNN
((standby))
=============================================
```

-- NOTES --

1. CCIR 476-3 defines MODE-B with two parts: COLLECTIVE (broadcast to all stations) and SELECTIVE (broadcast to one or a group of station(s)). I refer to MODE-B COLLECTIVE as MODE-Bc and MODE-B SELECTIVE as MODE-Bs. Throughout this document, terminology from CCIR Rec 476-3 has been used, where appropriate, to reduce confusion. A major

change Is the use of the CCIR 476 term "call" signal rather than the more common SELCALL of AMTOR.

2. effectively on a newline because of the CR LF within each SECALL string.

3. Alternatively, the SELCALL-DETECT flag may be set when the Q-CALL processor detects the sequence: "Q" "wxyz" "SPACE" "Q" "wxyz", provided that the receiving station has user-selectable option of using the longer sequence.

4. Including the character that caused the DATA-OUTPUT flag to be set.

5. this would be a problem for systems that use external Q-CALL processors and ASCII terminals.

6. this is often done using MODE-Bc to tell party B what party C's "call" signals are.

7. Some CCIR 476 units wait until a CR or LF is detected in MODE-Bc before outputting characters to the teleprinter.

PACKET RADIO SOFTWARE APPROACH - 1984 ONWARDS

Robert M. Richardson, W4UCH
22 North Lake Drive
Chautauqua Lake, N Y 34722

ABSTRACT:

The future of the software approach is 'crystal balled' by the author of 'Synchronous Packet Radio - The Software Approach' (Vols. 1, 2, & 3) and 'The Gunnplexer Cookbook - A 10 GHz Microwave Primer.' Topics discussed include:

- Bringing the software approach to the top of the learning curve.
- New generation microprocessors influence on the software approach.
- Low power requirement applications of the software approach to remote terrestrial repeaters and spacecraft.
- Implementing the software approach on very low-cost microcomputers.
- Future packet predictions year 2000 AD.

APPROACHING THE TOP OF THE LEARNING CURVE:

In 1982 the software approach was at the 35% level of the typical learning curve. Here in early 1984, the software approach is about at the 85% level of the learning curve.

With real-time zero insertion in the transmit mode, real-time synchronous to parallel byte conversion in the receive mode, and virtual real-time CRC generation and checking already accomplished, the last remaining bridge to cross is concurrent keyboard message input while in the receive mode decoding incoming synchronous packets.

The voluntary constraint of insisting that the software approach work on a first generation, personal microcomputer such as the circa 1976 Model I TRS-80 without hardware modification, ruled out the possibility of using the Z-80's interrupt modes. Had this been possible, the last bridge to cross would have been an easy one.

There is yet another approach to concurrent keyboard input while in the receive mode that the author used in 'Advanced Baudot Radio Teletype for the TRS-80' that was written in 1981. That is, using the idle countdown time between each mark and space sample to accomplish the keyboard scan and keyboard input processing. Using 100 speed Baudot TTY, this is indeed an easy programming task. At 1200 baud and up packet, it is a fascinating challenge. This is the approach we may use later this year.

NEW GENERATION MICROPROCESSORS:

With the introduction this year of the new Zilog Z-800 8/16 bit microprocessor starting off with a 10 MHz clock and planned growth to a 25 MHz clock in a year or so, the software approach's current baud rate limitation of 1200/2400 baud (and 4800 baud using a 4 MHz clock) will be increased considerably.

A 10 MHz clock Z-800 with its internal 256 byte RAM cache memory (even with relatively slow ancillary chips) will easily run 9600 baud synchronous packet using the software approach. Considerably higher speeds will be possible as the Z-800 matures.

New generation Very High Speed Integrated Circuit (VHSIC) chips developed under the Department of Defense' VHSIC program are now beginning to come off the production lines. These digital processing chips operating in the GHz region, many of which are gallium arsenide medium scale integrated circuit chips, will extend the speed of the software approach to regions undreamed of today. Truly, the sky appears to be the limit.

LOW POWER REQUIREMENT APPLICATIONS:

The software approach reduces the chip count requirement of a packet operating system many fold. The entire terminal node controller (TNC) is eliminated which includes the TNC's microprocessor, SDLC/HDLC controller, vast EPROM, dynamic RAM, UART, and considerable numbers of ancillary support chips.

The power supply requirement of the software approach is significantly less than that of the packet approach using a dedicated terminal node controller. Two obvious applications for a low power drain synchronous packet system are:

1. Remote packet repeaters that utilize solar cells, wind generators, fuel cells, or fueled generators for local Power (forgot water wheels/turbines),
2. Spacecraft repeaters (about as remote as you can get) which rely solely on solar cells for power.

By using the new CMOS version of the Z-80 microprocessor, CMOS dynamic RAM, CMOS

ROM to hold the software approach program, and CMOS ancillary chips, the power drain of a remote packet repeater would be determined almost solely by the repeater's transmitter RF output requirements.

If you have a very tight power budget to work with, we suggest that it might be wise to consider the software approach. Though not yet available in the CMOS version, the GLB Electronics PK-1 packet radio controller, which uses the software approach in EPROM for both the Vancouver and AX.25 protocols, draws a total of only 200 milliamps at 12 volts DC. CMOS could reduce the power drain 1 to 2 orders of magnitude,

SOFTWARE APPROACH ON LOW-COST MICROS:

Should be an easy task for most experienced assembly language programmers using any one of the popular cross assemblers on the market today. Most all of the low-cost micros have at least one port already decoded for cassette input and output.

The software approach only requires a single port and if only a single bit is available (some very low-cost micros require that the user manually turn the cassette on and off), the software approach may still be implemented if the user is willing to manually switch from transmit to receive and vice versa.

Low-cost micros come with varying amounts of memory, Most that we know of can be expanded to a minimum of 16K and include some version of Basic in ROM. Many can be expanded up to the 48K memory level.

16K of RAM memory is about the minimum that the 'economy' version of the software approach would require for home station operation. If only the automatic forwarding function of AX.25 is used (strictly for packet repeater operation), then the program could be probably shoe-horned down to about 4K of memory by a truly 'tight code' programmer.

The economy version would do away with many of the niceties and convenience options that the 48K MEM disk I/O version offers, yet would be adequate for the newcomer with a modest budget. Further, this very low-cost approach would probably lure many newcomers into the packet fold.

We are NOT suggesting that the 'chiclet' key type of micro be used for packet, but rather the next level of low-cost micro, at least with a normal size typewriter keyboard might be implemented with the software approach loading into 16K of memory via cassette.

We encourage you enterprising packeteers out there in amateur radio land to have a go at this worthwhile project.

Just imagine tens of thousands of low-cost microcomputer sows' ears magically turned overnight into AX.25 silk purses by your stalwart programming prowess. It would be a giant step forward for packet. If you do not choose to do it, there are some commercial firms who will.

CONCLUSION:

Seriously, the future of packet radio (which is tomorrow) has room for all varieties of radio amateurs whether they are appliance operator inclined, of the moon bounce variety, or even quadrature phase shift keying oriented on 1296 MHz and up. Not only is there room for all kinds, packet radio needs all kinds to reach the level of acceptance it deserves in a timely fashion,

Packet radio in the year 2000 AD? Only 16 short years away. Here are our same of our 'if wishes were horses' packet predictions.

1. Fully authorized on all the low bands using 300/600 baud MSK. Synchronous packet totally replaces asynchronous Baudot, ASCII, and AMTOR.

2. VHF bands using 9600 baud and up MSK. 19.2K baud packet is the standard much like 1200 baud packet today.

3. Low altitude orbit amateur satellites a thing of the past (like predicting the demise of buggy whips after the automobile went into mass production),

4. Level/layer 3 packet fully implemented via terrestrial and satellite links. Hopefully, this will occur long before the year 2000 AD.

5. 2300 MHz amateur band serves as uplink to geo-stationary satellites with multiple access and intra-satellite packet forwarding capability. Down-link broad beam in 1215 MHz band and spot beam in 5650 MHz band or higher.

6. Other microwave amateur bands: 'Crystalmatic frequency stabilization" system allows 10 GHz and 22 GHz solid-state narrow-band packet communication systems to be used for amateur point to point packet communications. This is already available today using low power Gunnplexers.

7. The software approach? Still alive and well because it is a cost effective approach to amateur packet radio communications. The intellectual challenge is a side benefit for those who like to understand what they are doing and who like to climb mental mountains because they are there.

PACKET RADIO - THE 3RD GENERATION SOFTWARE APPROACH

AX. 25 PROTOCOL

Robert M. Richardson, W4UCH
22 North Lake Drive
Chautauqua Lake, N.Y. 14722

ABSTRACT:

The 3rd generation 'software approach@ to 1200 baud packet radio using the AX.25 protocol is described. This approach consists of software written in assembly language to replace the Tucson Amateur Packet Radio (TAPR) terminal node controller (TNC) which includes:

- the TNC's 68093 microprocessor.
- the TNC's costly SDLC/HDLC controller.
- the TNC's large 25K to 35K EPROM.
- the TNC's dynamic RAM.
- the TNC's RS232 UART
- the TNC's ancillary support chips.

The software approach also eliminates the need for an RS232 interface (approx. $100 cost) on the host microcomputer which may be either a Model I or Model III TRS-80. The RS232 interface is replaced by a low cost port zero encoder/decoder (parts cost approx. $15) which is used to interface the microcomputer to a home brew modem (parts cost approx. $15) which may use the low cost EXAR 2206/2211 AFSK modulator and demodulator chips that are used in both the Vancouver and TAPR modems.

A more sophisticated modem of the users choice for noise-prone and fade-prone circuits such as OSCAR 10 may be required for satisfactory weak-signal operation, though the author regularly and reliably is able to work the Toronto, Canada area packet repeater, VE3PKT, some 110+ miles distant.

A number of major improvements for the 3rd generation packet radio software approach which are included in Volume 2, 'Synchronous Packet Radio Using The Software Approach - AX.25 Protocol,' are described in detail.

INTRODUCTION:

Just as the TAPR terminal node controller has undergone a number of iterations and improvements since its inception, the 'software approach' has followed a similar course. Looking at a typical exponential growth/learning curve, 1984's software approach is about 85% up the vertical scale and approaching the knee of the curve whereas the software packet program written in 1982 was at the 33% level. This decided improvement was largely motivated by the disclosure in 1983 to the public at large, of the brilliant AX.25 protocol by Terry Fox, WB4JFI et al at the Second ARRL Amateur Radio Computer Networking Conference. The AX.25 protocol is to packet, what SSB was to amateur radio communication techniques in the mid-1950's; i.e., not revolutionary, but a giant evolutionary step forward. We doff our collective hats to the many authors of the excellllent AX. 25 protocol.

The 3rd generation software approach has a significant number of improvements over the 1st generation that was presented in Volume 1, 'Synchronous Packet Using The Software Approach - Vancouver Protocol.' These improvements are:

1. Receive mode synchronous to parallel byte conversion is done in real-time.
2. Automatic; AX.25 repeater if your call+ SSID in repeater segment address field
3. CRC generation and checking is done in virtual real-time = 27 times FASTER.
4, AUTO connect mode option for unattended operation with T2 timer auto reset.
5. FORMAT on/off option for receive video recognizes or ignores C/R's and L/F's.
6. Multi-frame packets are input from the keyboard same as single frame packets.
7. Information field length may be set from the menu from 1 to 256 bytes.
8. Frames per packet may be set from the menu from 1 to 7 frames.
9. NOW connected mode displays and stores only the information field each frame.
10. NOT connected mode displays and stores everything except flags and CRC bytes.
11. Disk I/O from within the program.

Here is a rundown of major improvements.

A. REAL-TIME SYNCHRONOUS BIT CONVERSION:

In Volume 1, received packets were stored in memory using the byte per received bit approach. This was a great teaching tool as it allowed the user to visualize the SDLC received bit pattern a full page of memory (1024 bytes per page) at a time using the program's edit/modify mode. Each and every received bit, flags, data bits, and zero insertion bits were there to be seen. Some times a picture is worth a thousand words and it was quite useful for the newcomer to synchronous packet radio to be able to see the

brilliant IBM synchronous data link control concept in action.

So much for the pluses of this approach. Its disadvantages were that it took precious time to decode tne data after the packet had been received and more importantly ate up memory faster than a hungry bear. The time factor was not noticeable with single frame packets, but was measurable when multi-frame packets of maximum length were received. The voluminous memory requirement for the byte per received bit storage was this approach's major detriment.

Along comes Sir Galahad, ne Gil Boelke-W2EUP, on his white charger to rescue Volume 2 from the memory monster. Not only does W2EUP's superb real-time serial synchronous bit to parallel decimal byte conversion subroutine solve the memory problem, but it also eliminates the measurable time delay for decoding long multi-frame packets.

The author's software digital phase-locked loop (DPLL) used in Volume 1, was again used in Volume 2 with only cosmetic changes. It was an old trusty/reliable friend and allowed the user to copy 1200 baud packets whose timing was off as much as 10 percent from the norm. It is somewhat analagous to the hardware approach used by the Intel 8273 dedicated SDLC controller MSI chip. Figure 1 illustrates two, bit time periods where there was a change from space to mark (mark and space are used only as colloquial terms since SDLC/HDLC are only interested in the relative change and not the absolute value),

Each 1200 baud 833.333 microsecond bit time is divided into quadrants with each quadrant testing for a change of state (mark or space) of the incoming serial synchronous data bit. Ideally, all transitions from mark to space or vice versa, will occur exactly between quadrants 2 and 3, so that the bit sample taken after time 4 is exactly at the dead-center of each incoming bit time. Since our software DPLL is somewhat less than ideal/perfect, it adjusts the variable time 4 countdown value so that the average bit transition is usually between time 2 and time 3. If it occurs during time 1 or time 4 a much larger correction is made to time 4 to bring the sample time back to near dead-center again.

All bit processing is done by the program between time 4 and time 1. The bit processing time is less than 10% of the total 833.333 microsecond bit time period so has little or no effect on the DPLL as long as each processing time period is exactly the same. Equalizing time delays in the processing routine are used to insure that the processing time period is exactly the same. Equalizing time delays in the processing routine are used to insure that the processing time period remains constant.

The DPLL's fixed time constants of TYME 1, 2, and 3 with values of 23 decimal are for the Model I TRS-80. The Model III with its slightly faster clock uses decimal 28 for TYME 1, 2, and 3. The DPLL subroutine's calculated TYME 4 countdown correction values are the same for both Models.

Figure 3 is the commented source code for the 1200 baud real-time synchronous to parallel decimal conversion, most of which from lines 900 to 1880 were generously contributed by W2EUP. The author's DPLL begins at the label TYME in line 1880 and runs through the end of this subroutine. Fig. 3 starts off with MODE which is the beginning of the receive mode subroutine. All the folderol before line 900 are simply the cues to tell you what the program has done automaticaly (if in the NOW CONNECTED mode of operation), such as displaying <CONNECTION ACK> on video if the program was in the AUTO ON mode, and so forth.

In the receive mode, the program continually cycles between NEWONE in line 690 and line 840/860 while looking for a valid (mark or space) change in the input from the EXAR 2211 demodulator via port zero. When a change is found, the program progresses to FL1 where it searches for the first opening flag. If the DCD (data carrier detect) from the EXAR 2211 drops before a flag is found, it starts all over again at BEFOR1.

Once an opening flag has been found, it proceeds to FL2 where further opening flags are ignored as this subroutine is searching for the first non-flag data byte in the frame. Again, if DCD drops it starts all over again at BEFOR1. When the first non-flag byte of the first frame is assembled, line 1270 jumps of to line 1600. The IN1 subroutine is the work horse of this real-time receive mode decoding section.

Only the first flag that is decoded by FL1 is stored at 40959 in memory. Decoded packet data bytes are stored from 40960 on up in memory by the IN1 subroutine. All converted bytes except frame ending flags are stored here for each packet. Each frame's ending flag location is stored sequentially in memory beginning at STORE.

After the entire packet has been decoded in real-time, IN1 exits to the MOVEM subroutine that is not shown in Figure 3 as it is too lengthy for this conference paper. MOVEM's function is determined solely by the mode the operator has selected; i.e., NOW connected or NOT connected.

B. AUTOMATIC REPEATER + NOW/NOT CONNECT:

In the NOW connected mode of operation each frame is CRC checked and if ok, the repeater segment of the address field then tested. If it contains your call letters and SSID, then the repeated bit is set for each frame, the CRC recalculated for each frame, and the total packet retransmitted. As such, your packet station serves as an automatic repeater. Video will display <FORWARDING> when this function is used. If the automatic repeater function is not called, the program then tests the other station's and your call letters + SSIDs (and repeater call letters + repeated bit where applicable) and if ok, sequentially tests each frame's control byte to determine the function.

Let's assume it was an information frame. Since you know who you are connected to, (the other station's call letters are displayed by the program in the 1st three right hand columns of Figure 2's main menu in both the auto and non-auto modes), ONLY the information field of each frame is displayed on video and stored in high memory. The ACK is then transmitted automatically while the video display remains in the receive mode. See Figure 2 for the main and shift menus.

In the NOT connected mode, everything except the flags and each frame's CRC bytes are displayed on video. The call letters and repeater if used, of the address field are right shifted one bit so as to display their real ASCII values. If you selected the NOW FORMAT option from the main menu, all carriage returns and line feeds are recognized and acted upon on the video display, If NOT FORMAT, they are ignored and the TRS-80 symbols for ASCII 13 and 10 displayed. NOW or NOT format may be used in either the NOW or NOT connected modes.

Intentionally, there is no CRC check of each frame in the NOT connected mode as we wish to see everything the EXAR 2211 is capable of demodulating, good and bad, up to 4K bytes in length per packet. Simultaneously with the video display function, all received bytes are stored sequentially in high memory beginning at 53248 decimal. Each received packet with CRC bytes may be inspected a full 1024 bvte page at a time by going to the edit/modify mode via press M from the main menu to go to 40960 in mid-memory. Press ENTER to move up a page or the MINUS key to move down a page. 40960+ is re-used bv each received packet. To inspect everything sequentially received so far (up to 12 pages = 12,288 bytes) except flags and CRC bytes, press shift M to take you to 53248+ in memory and then page up or down in memory as you wish. The BREAK key will return you to the main menu from the edit/modify mode.

C. HI-SPEED CRC USING THE BYTE-WISE LOOK-UP TABLE APPROACH SUGGESTED BY PEREZ:

Volume 1's software CRC generation and checking subroutines emulated the hardware approach used by IBM and the other SDLC controller chip manfacuturers. By that we mean the software aprroach emulated the same multi-shift register format and derived the CRC value on a bit by bit basis. It worked very well thank you, but it ate up precious time, especially with long multi-frame packets.

One approach we tried was to do the transmit mode CRC generation in real-time while the frame was actually being sent, just as the Intel 8273 SDLC controller chip does it and just as this program does the zero-insertion in real-time. It worked, but it solved the wrong problem. The real problem lay in the receive mode CRC checking time delay that was measurable when maximum length multi-frame packets were being received.

Much like Sir Galahad saving the SDLC maiden from the memory monster, along comes Sir Lancelot, ne Aram Perez, and saves the CRC damsel from the time eating dragon. The June '83 issue of IEEE Micro Journal had a fascinating paper by Aram Perez that covered his 'byte-wise' CRC look-up table approach for the CRC16 (Bisync) polynominal. Without too much difficulty we were able to generate a look-up table for the IBM SDLC polynomial used by the AX.25 and Vancouver protocols.

The results? An incredible 27 times SPEED-UP of both CRC checking and generation compared with Volume 1 of the software approach. The majority of this section and its subroutine is courtesy of Mr. Perez' excellent paper.

The CRC we will cover will detect:

- all one or two bit errors.
- all odd number of bit errors.
- all burst errors less than or equal to the the degree of the polynomial used.
- most burst errors greater than the degree of the polynomial used.

What this adds up to in AX.25 protocol is a probability in the vicinity of 10 to umpteenth power, that if the CRC tests ok, the received frame that was CRC checked is correct and identical to that which was transmitted. If it is good enough for banks to transfer funds by electronic mail (it is), it should be good enough for us.

HERE IS HOW IT WORKS:

In a protocol utilizing the cyclic redundancy check, the message to be transmitted between the last opening flag and the closing flag in each frame is considered to be a binary polynomial M(X). It is first multiplied by X to the K power and then divided by an arbitrary generator polynomial G(X) of degree K. This yields a quotient Q(X) and a remainder R(X) divided by G(X). All arithmetic is done in modulo 2; i.e., the results of subtraction are equal to the results of addition. The

equation looks like this:

$$\frac{x\,M(X)}{G(X)} = Q(X) \oplus \frac{R(X)}{G(X)}$$

The $\oplus$ sign signifies addition in modulo 2 arithmetic. Simplifying this equation yields:

$$x\,M(X) \oplus R(X) = Q(X)G(X)$$

Where R(X) will always be of degree K or less. The CRC algorithm calculates R(X) and tacks these 2 bytes onto the end of the frame to be transmitted. Since as illustrated above, x M(X) $\oplus$ R(X) does indeed equal Q(X)G(X), the original message with the 2 byte CRC tacked on will be evenly divisible by G(X), IF and only IF no bits were erroneously received. Using the IBM SDLC (CCITT) polynomial as shown below, the remainder will always be 61624 decimal IF the frame was received correctly.

IBM SDLC AND BISYNC GENERATOR POLYNOMIALS

NOTE the [figure = exponentiation

IBM SLDC (CCITT)	X[16+X[12+X[5+1
SDLC REVERSE	X[16+X[11+X[4+1
CRC1 6(BISYNC)	X[16+X[15+X[2+1
CRC16 REVERSE	X[16+X[14+X[1+1

The reverse polynomials are the same as their big brothers, except taken in reverse order. Since the rather simple CRC arithmetic is done in modulo 2, it is quite easily implemented by the MSI chips used by both Vancouver and TAPR TNC boards. The former using the Intel 8273 MSI chip and the latter using the Western Digital 1933/1935 MSI chip.

One of the drawbacks to using the hardware rather than the software approach is that the user never knows what the CRC value is that he/she transmitted or received. Some packet operators could care less, but then again, some radio amateurs prefer to fully understand what they are doing.

This program allows you to read out exactly what the generated and received CRC values are for every packet that is transmitted or received by using the edit/modify mode.

Unfortunately there is no such thing as 'free lunch.' The price we have to pay for this extremely FAST CRC subroutine is a modest bit of memory for the 512 byte lookup table. Nevertheless, it is a small price to pay for the near 'speed of light' swiftness gained. Again, this approach is 27 times faster than the bit by bit CRC routine used in Volume 1.

Both received frame CRC checking and transmit frame CRC generation are each quite simple using Aram Perez' byte-wise approach modified for IBM SDLC (CCITT) polynomial. Let's look at the transmit mode CRC generation first.

All frames to be transmitted are first moved to MEM location 43008 + a frame at a time, then the CRC is generated, and inserted. For multi-frame packets, a frame is moved, the CRC generated for that frame and inserted, and then the next frame moved, CRC generated and inserted, etc. This only requires milliseconds of real-time.

The memory location denoted by the label ENDCRC always contains the generated CRC value of the packet just transmitted IF it was a single frame packet or the generated CRC value of the last frame transmitted if it was a multi-frame packet. If the current packet being transmitted is a single frame info packet and the program in the NOT connected mode, the CRC value in decimal will be displayed on the top line of video, and the packet immediately below it while it is being transmitted.

Why bother with displaying the CRC decimal value in the unconnected mode?

Only for convenience. Sometimes it can be very useful for the station at the other end who is trouble shooting his/her receive mode system. Even the hardware approach using the Western Digital WD-1933 or Intel 8273 SDLC chips can on occasion have problems with its real-time CRC. Some of the early SDLC controller chips exhibited this type of problem.

Figure 4 starts off with the commented source code for generating the two IBM SDLC CRC bytes for each frame to be transmitted. Almost the same routine is used for testing the CRC value of each incoming frame of each packet. See lines 870 through lines 990 of this subroutine for the receive mode CRC check. For either transmitted or received frames, this CRC function is accomplished virtually in real-time.

D. TRANSMITTING MULTI-FRAME PACKETS:

Data for the information fields of all multi-frame packets originates in low memory beginning at 17408 decimal. 12288 LO-MEM bytes are reserved here for the automatic multi-frame transmit function. Data may be input directly from the keyboard by pressing shift L to go to 17408 in LO-MEM in the edit/modify mode and then typing away till done, or data may be input from disk without leaving the program.

Referring to Figure 2's main menu, the operator presses G to input the number of frames per packet 1 - 7, and then presses N to input the information field length of 1 to 256 bytes per frame. Actually, any info field length up to 2000 bytes may be specified for use between agreeing packeteers if a reliable path is available. Now, press E to commence the LO-MEM multi-frame transmit function.

In NOW connected mode, the program

will calculate the number of frames to be transmitted, divide them by the number of frames per packet specified, calculate the total number of packets to be transmitted, calculate the number of frames in the last packet, and calculate the number of bytes in the last frame of the last packet. It will then begin sending them automatically. While they are being transmitted, the top line of video will display the remaining number of frames to be transmitted, and up to the first 15 lines of the packet being sent.

After the packet is transmitted, the program will switch to the receive mode and display <OK> if the acknowledgment was correctly received, or <RESENDING> if it was not received correctly or the T1 resend timer times out. Assuming that the ACK was correctly received, it will then assemble and transmit the next packet. The total assembly time for each multi-frame packet including CRC'ing each frame, is measured in milliseconds. This process will continue automatically till all LO-MEM data has been transmitted and acknowledged.

In the NOT connected mode, the operation is almost identical to that just described, except the operator must press the E key from the main menu to sequence and then transmit each packet till all LO-MEM has been transmitted, as ACK's will obviously not be received. This function is seldom if ever used in the NOT connected mode and was included only to satiate one of our rather unique BETA testers who gets his jollies from sending long multi-frame packets in this mode.

Figure 5 is the commented source code for the multi-frame transmit mode subroutine. It is easy to follow when one understands how the regular registers, alternate registers, and stack are used from SEND7 onward.

REGULAR REGISTERS:
A = parallel byte from memory
BC= time delay routine in SN1
D = parallel byte value in SN1
E = bits per byte counter SN1
HL = JP (HL) countdown jump SN1
IX = unused
IY = xmit byte memory location

ALTERNATE REGISTERS:
A = unused
B = frames in the current packet
C = last frame last pack pointer
DE = last frame last packet length
HL = frame length except for last

STACK IN SEND7:
Bytes remaining to send in frame

The SEND7 subroutine in Figure 5 is not really a sticky wicket if one realizes that the program always sets alternate C register to 1 more than B register, except for the last packet being transmitted from LO-MEM. As such, it never jumps to KYBD4B except for the last frame of the last packet. For the last multi-frame packet only, alternate C and alternate B are set to one less than the number of frames to transmit in this final packet. When the next to last frame of this last packet has been transmitted, alternate C is decremented to zero, so jumps off to KYBD4B that pushes alternate DE on the stack which is the length of the final frame of the last packet.

The SN1 and SN1A subroutines in Figure 5 do the yeoman job of converting the parallel decimal byte to the synchronous 1200 baud serial bit that is output via port zero. SN1A is used for 126 decimal flags that do not utilize zero-insertion, and SN1 is called for data bytes between flags that may require zero-insertion.

E. DISK I/O FROM WITHIN PACKET PROGRAM:

At first glance appears as easy as falling off a log. Always be suspicious of easy logs in this game. On second glance, when one realizes that virtually all of RAM memory from 17408 to 28672 is used by the TRS-80 for disk subroutines, and this is the area where the software approach stores long data from the keyboard or disk to be transmitted in multi-frame packets, it becomes apparent that both the packet data and disk subroutines cannot occupy the same memory at the same time.

One simple solution is to leave the packet program, do the disk I/O functions desired, return to the packet program, clear out low memory, and resume packeteering. Though simple and easy to accomplish, it is a decided inconvenience and time consuming approach for the operator.

What we desired was having our cake and eating it too; i.e., having the write to disk and read from disk functions within the software approach program, while at almost the same time being able to use low memory for storing long data to be transmitted in multi-frame packets.

The solution to this apparent paradox was to save the TRS-80's minimum disk operating system (system 1) in mid-memory and write our own disk I/O subroutine that this section delineates. Our disk I/O subroutine requires only 1859 bytes of memory and serves three purposes:

1. Volume 2 is a teaching textbook as well as a working AX.25 program. As such, it familiarizes the reader with writing direct disk I/O subroutines.
2. Allows disk I/O without leaving the packet program.
3. Provides the basis for Volume 3's automatic-unattended disk access by another packet station. In essence, it is a mini-version of a computer bulletin board system with the SEND, SAVE, LIST,

and HELP commands sent via packet.

Figure 2% SHIFT menu illustrates the 3 commands used for the disk I/O functions from within the software approach program. Shift R loads a disk file of up to 12K bytes in length to high memory (53248 up) and shift D moves it low memory for multi-frame packet transmission. Shift Q saves up to 12K bytes of high memory in a disk file of whatever name the operator wishes to give it. The high memory data may be either input from the keyboard using the edit/modify mode, or conversely may be received packets the operator wishes to save on disk.

Figure 6 is the commented source code for this subroutine which is largely self-explanatory. It works quite well with the Model I TRS-80 and on a maybe basis for the Model III TRS-80 depending upon which version of ROM the user's system has installed.

F. REAL-TIME EDIT/MODIFY/MONITOR MODE FOR COMPUTER NETWORKING PROGRAMS:

Whether the software or hardware approach to packet radio is used, we have found that an in-program (within the terminal interface program TIP) subroutine that allows instant access to the computer's 64K bytes of memory, 1024 bytes per page displayed on video, is a useful adjunct to the packet operator% tool kit.

Memory may be reviewed in the edit mode and modified in the modify mode if desired. If the operator wishes to save the modified TIP it may be dumped to disk thus eliminating the time consuming requirement of exiting the TIP program, loading the TIP source code into an Editor/Assembler, modifying the source code, assembling the program, and then writing it on disk.

In addition to the edit/modify/monitor functions, this subroutine serves as the keyboard input subroutine for typing packet messages into low memory beginning at 17408. Up to 12 pages, 1024 bytes per page, may be used by enthusiastic typists. A carriage return followed by a line feed is input by pressing ENTER, End of message delimiters, 128 decimal, are input by pressing shift zero.

The short 866 byte subroutine that performs the edit/modify mode functions is illustrated in Figure 7 which is the commented source code.

The edit/modify program may be considered a subroutine if you wish, but it is truly a stand alone program that may be appended to any variety of software where the user wishes to access to all 64K bvtes of memory WITHOUT leaving the program. Depending on the ROM/RAM varieties in the particular computer, the user may not only review, but actively modify anywhere from 48K to 64K of memory while the program is up and running.

EDIT/MODIFY PROGRAM ENTRY POINTS:

There are 3 entry points to save the user the trouble of having to page too far through memory. They may be called from the TIP program's main menu in Figure 2 by:

1. Press M to go to the 1024 byte page of beginning in mid-memory at 40960 decimal.
2. Pressing SHIFT M from the menu will display-the 1024 byte memory paqe beginning at 53248 in high memory.
3. SHIFT L from the menu will display the 1024 byte memory page beginning at 17408 in low memory.

We will assume you pressed M from the TIP menu which takes us to memory location 38912 that is in line 5240 of Figure 7's source code program. Had you pressed SHIFT M or SHIFT L, then the HL register would have been loaded with 53248 or 17408, respectively and the jump made to 38915 in MEM that is in line 5250.

The rest of the subroutine in Figure 7's commented source code is largely self-explanatory.

The edit/modify/monitor in-program subroutine is a useful tool for the packeteer. It is elegant in its simplicity, yet a very POWERFUL tool. By all means modify it to suit your own operating habits and fancy. If you are used to using memory modifier and/or monitor programs such as SUPERZAP, DEBUG, ZAPSIT, etc., you may abandon them for this short in-program subroutine once you become accustomed to using it.

A new version of the edit/modify subroutine using a number of the Electric Pencil (tm) word processing program commands for keyboard input of packet messages may be implemented later this year.

CONCLUSION:

First, a personal note. Writing the 'software approach' for both Volumes 1 and 2 was great fun and very gratifying.

why?

Because so many experienced packeteers told us it could not be done using a modestly priced 2 MHz ballpark crystal clock Model I or Model III TRS-80 microcomputer. Actually, most any computer with a 1 MHz or faster clock should be able to handle 1200 baud synchronous packet using the software approach. The Model I or Model III TRS-80 is quite capable of running 2400 baud packet using this program if the timing constants are carefully trimmed and adjusted.

With the new Ziloq Z-800 microprocessor and its extremely fast clock, (and internal cache memory), the software approach may be extended to 9600 baud and well beyond.

Want to dig deeper? If so, try Volume 1 or 2 of 'Synchronous Packet Radio Using The Software Approach.'

Vol. 1 - Vancouver Protocol is $22 ppd and Vol. 2 - AX.25 Protocol also $22 ppd. If you want the programs on disk in addition to the book which is required for instructions to personalize the disk, specify Model I or Model III TRS-80. The disk programs are an additional $29 ppd. U.S. phone orders are welcome during business hours at (716)-753-2654 or you may order from:

Richcraft Engineering Ltd.
#1 Wahmeda Industrial Park
Chautauqua, New York 14722

Do not want to dig deeper? Then we highly recommend to you the Tucson Amateur Packet Radio terminal node controller. It is a highly efficient, very professional, and first-rate kit. It is available for $252 which is about one half the price were it produced by a profit making enterprise that most likely would not do as thorough a job as TAPR.

We salute TAPR and all those who have contributed to the development of its TNC, for an outstanding service to amateur radio.

REFERENCES:

Proceedings - 2nd ARRL Amateur Radio Computer Networking Conference
AX.25 Protocol pp 4 -14
by Terry Fox, WB4JFI

Proceedings : 2nd ARRL Amateur Radio Computer Networking Conference
Link Level Address Mechanisms pp 47-49
by Henry S. Magnuski, KA6M

IEEE Micro Journal - June 1983 issue
Byte-Wise CRC Calculations pp 40 - 50
by Aram Perez

QST - February 1984 issue
'On Line' Column pp 77
by Stan Horzepa, WA1LOU

Gunnplexer Cookbook - A 10 GHz Microwave Primer 335 pp
by Bob Richardson, W4UCH

Advanced Baudot RTTY for the TRS-80
Vol 5 Disassembled Handbook 223 pp
by Bob Richardson, W4UCH

AX.25 Protocol Modifications/Update personal communication
by Paul L. Rinaldo, W4RI

REFERENCES continued:

Z-800 Micro-P Product Specification
Zilog, Inc.
1315 Dell Avenue
Campbell, CA 95008

FIGURE 1

1200 BAUD SOFTWARE DIGITAL PHASE-LOCKED LOOP QUADRANTS

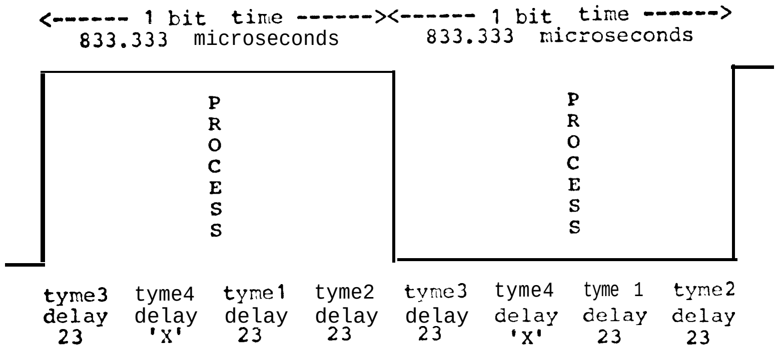


FIGURE 2: ENTER OPTION DESIRED ? _

CHANGE ADDRESSEE CALL LTRS = A	W2EUP CONNECT REQUEST CQ = B
MOT CONNECTED TOGGLE = C	W2EUP DISCONNECT REQUEST = D
SEND PACKETS FRO!: LO-MEM = E	W2EUP CONNECT ACKNOWLEDGE = F
INPUT FRAMES/PACKET LO-MEM = G	THIS IS AX.25 PROTOCOL = H
BACKOFF DELAY TOGGLE OFF = I	AUTO CONNECT TOGGLE OFF = J
NOW IN UPPER CASE MODIFY = K	W2EUP - GIL BOELKE MESSAGE = L
DISPLAY/EDIT MEMORY PAGE = M	SET INFO FIELD LOMEM PACKS = N
NOW FORMAT VIDEO TOGGLE = 0	QUICK BROWN FOX MESSAGE = P
VIA WA2EGW/R REPEATER ON = Q	SET OPENING FLAG LENGTH = R
CHANGE REPEATER CALL LTRS = S	INPUT/XMIT NORMAL INFO = V & T
CLEAR NON-PGM MEM 17K-62K = U	INPUT/XMIT UNNUMB INFO = V & W
ABORT LOW-MEM PAK SEQUENCE = X	NOT CONEK TO OWN STATION = Y
SHIFT MENU = 1	SET RE-TRY IN CONNECT MODE = 2
SEND WAIT REQUEST (RNR) = 3	SEND CLEAR WAIT (RR) = 4
(not shown) :	(not shown):
DISCONNECTED MODE = 5	FRMR FRAME REJECT = 6

SHIFT MENU 3 _

XMIT 40960 Up CONTINUOUSLY = A	BOOT DOS READY = B
LOAD MID-ME": ASCII UUUUUU = C	MOVE HI-MEM TO LOW-MEMORY = D
EDIT/MODIFY INSTRUCTIONS = E	CHANGE RECEIVE DPLL BASE # = F
TRANSMIT EXTERNALLY ONLY = G	TRANSMIT TO HI-MEMORY ONLY = H
SEND MORSE: I.D. = I	SEND SEQUENTIAL ACKS = J
CAUTION ** RESTORE DOS ** = K	DISPLAY LOW MEMORY @ 17403 = L
DISPLAY RECV PACKS @ 53248 = M	RESTORE PROGRAM POINTERS = N
OSCAR 10 CALL/HANDLE LIST = 0	MOVE PROGRAM TO LOW MEMORY = P
SAVE HI-MEM OIJ DISK = Q	LOAD DISK FILE TO HI-MEM = R
TRANSMIT BAUD RATE SELECT = S	TEST OTHER STATION STATUS = T
CLEAR HI-MEMORY 53248 + = U	SEND MORSE FROM KEYBOARD = V
RECEIVE AX.25 PROTOCOL = W	RECV VANCOUVER NOT CONNECT = X
NORMAL DISPLAY - NOT DPLL = Y	DISPLAY DPLL LAST QUADRANT = Z

NOTE: SPACE BAR IN RECEIVE MODE = RESEND LAST PACK

```
00100 ;                    FIGURE 3
00110
00120 ; RECEIVE MODE REAL-TIME SDLC/HDLC SERIAL SYNCHRONOUS
00130 ; DATA STREAM TO PARALLEL DECIMAL BYTE CONVERSION.
00140
00150 ; THE REGISTERS USED IN THIS RECEIVE MODE SUBROUTINE FROW
00160 ; LINE 900 ON ARE:      REGULAR REGISTERS
00170 ; A = USED + NEW PORT ZERO VALUE IN EACH DPLL QUADRANT
00180 ; F = USED THROUGHOUT
00190 ; B = DPLL COUNTDOWN VALUE FOR FIRST 3 DPLL QUADRANTS
00200 ; C = 8 BITS PER BYTE COUNTER
00210 ; D = CALCULATED DPLL COUNTDOWN VALUE FOR 4TH QUADRANT
00220 ; E = LAST PORT ZERO VALUE
00230 ; HL= MEM LOCATION TO STORE ENDING FLAG ADDRESS
00240 ; IX= ONLY FOR EQUALIZING TIME DELAYS; INC IX & DEC IX
00250 ; IY= UNUSED
00260 ;                    ALTERNATE REGISTERS
00270 ; A = UNUSED
00280 ; F = UNUSED
00290 ; B = RECEIVED PARALLEL BYTE WITH ZERO-DELETION
00300 ; c = RECEIVED PARALLEL BYTE WITHOUT ZERO-DELETION
00310 ; D = INCOMING BIT VALUE AT CENTER OF BIT TIME FRAME
00320 ; E = LAST BIT VALUE AT CENTER OF BIT TIME FRAME
00330 ; HL= MEM LOCATION TO STORE CONVERTED DECIMAL BYTE
00340
00350 ; THIS SUBROUTINE IS ENTERED IN LINE 440, 490, OR 500
00360 ; DEPENDING ON WHETHER RECEIVE MODE IS ENTERED FROM THE
00370 ; MAIN MENU, NOT CONNECTED MODE, OR NOW CONNECTED MODE.
00380
00390 ; THE SOFTWARE DIGITAL PHASE LUCKED LOOP (DPLL) IS AT THE
00400 ; END OF THIS SUBROUTINE IN LINES 1880 - 2230.
00410
00420 ; SIGNIFICANT RECEIVE MODE SUBROUTINES FROM VOLUME 2
00430
00440 MODE    LD     BC,6500      ;.01 SECOND DEBOUNCE
00450         CALL   060H         ;TIME DELAY SINCE THE
00460         LD     A,(14400)    ;CLEAR KEY IS USED TO
00470         CP     2            ;TOGGLE BETWEEN THE MENU
00480         JP     Z,MODE       ;AND RECEIVE MODE.
00490 MODE1   CALL   RESKCV       ;RESTORE RECEIVE VIDEO
00500 MODE1A  CALL   TESTSP       ;TEST SP FOR PGM ERRORS
00510         LD     A,(SIGN7)    ;DISPLAY ON VIDEO
00520         CP                  ;THAT A CONNECTION
00530         CALL   Z,CNRQ       ;ACKNOWLEDGE WAS SENT.
00540         LD     A,(SIGN6)    ; IF LONG DATA FROM LOMEM,
00550         CP     1            ;UP TO 12288 BYTES, THEN
00560         CALL   Z,SETIT      ;RESET POINTERS.
00570         LD     A,(SIGN5)    ;IF AX.25 STATUS REQUEST,
00580         CP     1            ;THEN DELAY 1 SECOND
00590         JP     Z,SPACK-10   ;BEFORE SENDING RR/RNR.
00600         LD     A,(SIGN4)    ;DISPLAY ON VIDEO
00610         CP                  ;THAT <DISCONNECT ACK>
00620         CALL   Z,DISCAK     ;WAS TRANSMITTED.
00630 BEFOR1  EXX                 ; SWAP ALTERNATE REGISTERS
00640         LD     HL,40959     ;MIDMEM TO STGRE PACKET
00650         LD     DE,0         ;INITIALIZE AT ZERO
00660         LD     BC,0         ;INITIALIZE AT ZERO
00670         EXX                 ;RESTORE REG. REGISTERS
00680         CALL   CLRMUL       ;CLEAR CLOSING FLAG STORE
```

```
00690 NEWONE  LD    A,(AUT)        ;AUTOMATIC CONNECT MODE ?
00700         CP    1              ;IF SO, AND CONNECTED, T2
00710         CALL  Z,TIMOUT       ;TIMES OUT 6 1/2 MINUTES.
00720         LD    A,(RTRY)       ;IN RE-TRY CONDITION ?
00730         CP    1              ;THEN ACTUATE T1 RE-TRY
00740         CALL  Z,TESTRY       ;TIMER BEFORE RESENDING.
00750         IN    A,(0)          ;EX-2211 OUTPUT PORT ZERO
00760         LD    D,A            ;SAVE IT IN 'D' REGISTER
00770         LD    A, (14400)     ;KEYBOARD PSUEDO MEMORY
00780         CP    2              ;CLEAR KEY PRESSED ?
00790         JP    Z,MENU0        ;IF SO, GOTO MAIN MENU
00800         CP    128            ;SPACE BAR PRESSED ?
00810         JP    Z,RSEND        ;IF SO, MANUAL RESEND.
00820         IN    A,(0)          ;EX-2211 OUTPUT PORT ZERO
00830         CP    D              ;ANY CHANGE SINCE LAST ?
00840         JP    Z,NEWONE       ; IF NOT, GO LOOK AGAIN
00850         BIT   0,A            ;DCD CARRIER DETECT ?
00860         JP    Z,NEWONE       ;NO 1200/2200 TONES
00870         LD    HL,STORE       ;END FLAG ADDRESS STORE
00880         LD    A,(DVAL)       ;DPLL COUNTDOWN VALUE
00890         LD    D,A            ;START OFF WITH NOMINAL
00900 FLG1    CALL  TYME           ;SOFTWARE DPLL LINE 1880
00910         INC   (IX)           ;EQUALIZING TIME DELAY
00920         DEC   (IX)           ;EQUALIZING TIME DELAY
00930         INC   (IX)           ;EQUALIZING TIME DELAY
00940         DEC   (IX)           ;EQUALIZING TIME DELAY
00950         BIT   0,A            ;DCD CARRIER DROPPED ?
00960         JP    Z,BEFOR1       ;THEN START OVER AGAIN
00970         EXX                  ;SWAP ALTERNATE REGISTERS
00980         LD    D,A            ;SAVE INCOMING BIT IN 'D'
00990         XOR   E              ;COMPARE WITH LAST ONE
01000         CPL                  ;DATA IN BIT 7
01010         LD    E,D            ;UPDATE E FOR NEXT ONE
01020         RLCA                 ;SHIFT INTO CARRY
01030         RR    C              ;INCOMING BIT PATTERN
01040         LD    A,C            ;SWAP FOR COMPARE
01050         CP    126            ;FOUND AN OPENING FLAG ?
01060         JP    Z,FLG2+31      ;IF SO, GOT0 LINE 1280
01070         EXX                  ;ELSE GO BACK TO FLG1
01080         JP    FLG1           ;START LOOKING AGAIN.
01090 FLG2    BIT   0,A            ;DCD CARRIER DROPPED ?
01100         JP    Z,BEFOR1       ;THEN START OVER AGAIN
01110         EXX                  ;SWAP ALTERNATE REGISTERS
01120         LD    D,A            ;INCOMING BIT VALUE TO D
01130         XOR   E              ;COMPARE WITH LAST ONE
01140         LD    E,D            ;UPDATE E FOR NEXT TIME
01150         CPL                  ;DATA IN BIT 7
01160         RLCA                 ;SHIFT INTO CARRY
01170         RR    B              ;INPUT DATA -
01180         RRCA                 ;ACCUMULATES HERE.
01190         RR    C              ;INCOMING BIT PATTERN
01200         LD    A,C            ;TEST IT
01210         CP    126            ;FOR ANOTHER OPENING FLAG
01220         JP    Z,FLG2+41      ;IF SO, JUMP TO LINE 1330
01230         LD    A,B            ;BUILT UP DATA VALUE
01240         EXX                  ;RESTORE REG. REGISTERS
01250         DEC   C              ;DECREMENT BIT COUNTER
01260         JP    NZ,FLG2+35     ;NOT ZERO, GET NEXT BIT
01270         JP    IN1A+1         ;1ST FRAME DATA GOTO 1600
01280         LD    (HL),A         ;STUFF 1ST FLAG HERE
01290         EXX                  ;RESTORE REG. REGISTERS
01300         LD    C,8            ;RESET BIT/BYTE COUNTER
01310         CALL  TYME           ;DIGITAL PHASE LOCK LOOP
01320         JP    FLG2           ;GO LOOK AGAIN
01330         INC   (IX)           ;EQUALIZING TIME DEALY
01340         DEC   (IX)           ;EQUALIZING TIME DELAY
01350         JP    FLG2+32        ;GO LOOK FOR NEXT BYTE
01360 IN1     BIT   0,A            ;PACKET TONES DROPPED ?
01370         JP    Z,MOVEM+1      ;IF SO, PROCESS IT.
01380         EXX                  ;SWAP ALTERNATE REGISTERS
01390         LD    D,A            ;INCOMING BIT VALUE TO D
01400         XOR   E              ;COMPARE WITH LAST ONE
01410         LD    E,D            ;UPDATE E FOR NEXT TIME
01420         CPL                  ;DATA IN BIT 7
01430         RLCA                 #SHIFT INTO CARRY
01440         RR    B              ;INPUT DATA BITS -
01450         RRCA                 ;ACCUMULATES HERE.
01460         RR    C              ;INCOMING BIT PATTERN
01470         LD    A,C            ;TEST IT
01480         CP    126            ;FOR A CLOSING FLAG ?
01490         JP    Z,FL1          ;IF SO, GOT0 LINE 760
01500         CP    254            ;PACKET TONES DROPPED ?
01510         JP    Z,MOVEM        ; IF SO, PROCESS IT
01520         AND   254            ;REMOVE BIT ZERO
01530         CP    124            ;0111110X PATTERN ?
01540         JP    Z,DELETE       ;IF SO, DO ZERO DELETION
01550         LD    A,B            ;BUILT UP DATA VALUE
01560         EXX                  ;RESTORE REG. REGISTERS
01570         DEC   C              ;DECREMENT BIT COUNTER
01580         JP    NZ,IN4         ;NOT ZERO, GET NEXT BIT
01590 IN1A    NOP                  ;SAVED FOR DPLL TESTING
01600         EXX                  ;SWAP ALTERNATE REGISTERS
01610         INC   HL             ;BYTE STASH MEM LOCATION
01620 IN2     LD    (HL),A         ;STASH IT IN MEMORY
01630         LD    A,H            ;TOO LONG A PACKET ?
01640         CP    176            ;OVER 4096 BYTES LONG ?
01650         JP    Z,MOVEM-3      ;IF SO, PROCESS IT
01660 IN3     EXX                  ;RESTORE REG. REGISTERS
01670         LD    C,8            ;RESET BITS/BYTE COUNTER
01680 IN4     CALL  TYME           ;DIGITAL PHASE LOCK LOOP
01690         JP    IN1            ;CONVERT INCOMING BIT
01700 FL1     PUSH  HL             ;GOT A CLOSING FLAG
01710         EXX                  ;RESTORE REG. REGISTERS
01720         POP   Bc             ;FLAG LOCATION MINUS ONE
01730         INC   Bc             #FLAG MEM LOCATION
01740         LD    (HL),C         ;STORE FLAG ADDRESS LSB
01750         INC   HL             ;NEXT STORE LOCATION
01760         LD    (HL),B         ;STORE FLAG ADDRESS MSB
01770         INC   HL             ;NEXT STORE LOCATION
01780         LD    A,144          ;OUT OF BOUNDS DUE TO -
01790         CP    H              ;RUN AWAY TNC ?
01800         JP    Z,MOVEM+1      ;IF SO, PROCESS IT
01810         JP    IN3+1          ;ELSE GO FOR NEXT ONE
01820 DELETE  RL    B              ;ZERO DELETION, SO -
01830         EXX                  ;BACKUP ALTERNATE B
01840         INC   (IX)           ;EQUALIZING
01850         DEC   (IX)           ;TIME DELAY.
01860         CALL  TYME           ;DIGITAL PHASE LOCK LOOP
```

```
01870         JP     IN1            ;CONVERT NEXT BIT
01880 TYME    LD     A,(14400)      ;ESCAPE IS CLEAR KEY
01890         CP     2              ;IF PRESSED GOTO -
01900         JP     Z,MENU0-1      ;MAIN MENU FOR INSTRUCTS.
01910         LD     B,23           ;MODEL I COUNTDOWN VALUE
01920 TYME1   DJNZ   TYME1          ;1ST QUADRANT COUNTDOWN
01930         IN     A,(0)          ;PORT ZERO VALUE TO 'A'
01940         CP     E              ;ANY CHANGE FROM LAST ?
01950         JP     NZ,DEC2        ;IF SO, GOT0 LINE 2120
01960         LD     B,23           ;MODEL I COUNDOWN VALUE
01970 TYME2   DJNZ   TYME2          ;2ND QUADRANT COUNTDOWN
01980         IN     A,(0)          ;PORT ZERO VALUE TO 'A'
01990         CP     E              ;ANY CHANGE FROM LAST ?
02000         JP     NZ,DEC1        ;IF SO, GOT0 LINE 2150
02010         LD     B,23           ;MODEL I COUNTDOWN VALUE
02020 TYME3   DJNZ   TYME3          ;3RD QUADRANT COUNTDOWN
02030         IN     A,(0)          ;PORT ZERO VALUE TO 'A'
02040         CP     E              ;ANY CHANGE FROM LAST ?
02050         JP     NZ,INC1        ;IF SO, GOT0 LINE 2180
02060         LD     B,D            ;ADJUSTED COUNTDOWN VALUE
02070 TYME4   DJNZ   TYME4          ; 4TH QUADRANT COUNTDOWN
02080         IN     A,(0)          ;PORT ZERO VALUE TO 'A'
02090         CP     E              ;ANY CHANGE FROM LAST ?
02100         JP     NZ,INC2        ;IF SO, GOT0 LINE 2210
02110         RET                   ;DPLL DONE. GO PROCESS IT
02120 DEC2    LD     E,A            ;SAVE NEW BIT IN 'E'
02130         LD     D,15           ;WAY TOO LATE, SO SHORTEN
02140         JP     TYME2-2        ;LAST QUAD COUNT A BUNCH.
02150 DEC1    LD     E,A            ;SAVE NEW BIT IN 'E'
02160         LD     D,20           ;TINY BIT TOO LATE, SO -
02170         JP     TYME3-2        ;SHORTEN LAST QUAD A BIT.
02180 INC1    LD     E,A            ;SAVE NEW BIT IN 'E'
02190         LD     D,24           ;TINY BIT TOO SOON, SO -
02200         JP     TYME4-2        ;LENGTHEN LAST QUAD A BIT
02210 INC2    LD     E,A            ;SAVE NEW BIT IN 'E'
02220         LD     D,29           ;WAY TOO SOON, LENGTHEN
02230         RET                   ;LAST QUADRANT A BUNCH.
02240 ; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
02250 ; END OF SYNCHRONOUS BIT TO PARALLEL BYTE CONVERSION VOL 2
```

```
00100 ;                      FIGURE 4
00110
00120 ; IBM SDLC CRC GENERATION AND CRC CHECKING SUBROUTINES
00130
00140 ; CRC1 AND CRC2 ARE FOR GENERATING THE 2 BYTE CRC VALUE
00150 ; FOR A FRAME OF (LENG1) BYTES IN LENGTH.  ADDREZ IS THE
00160 ; MEMORY LOCATION OF THE BEGINNING OF THE SINGLE FRAME
00170 ; PACKET TO BE TRANSMITTED.  MULTI-FRAME PACKETS USE A
00180 ; VARIABLE ADDREZ DEPENDING UPON WHERE EACH FRAME HAS
00190 ; BEEN SEQUENTIALLY MOVED IN MEMORY STARTING AT 43008.
00200
00210 ; RCRC BEGINNING IN LINE 870 TESTS THE RECEIVED CRC VALUE
00220 ; OF A FRAME STARTING AT (BGINIT) IN MEMORY WITH A TOTAL
00230 ; LENGTH OF 'BC' REGISTER BYTES.  MULTI-FRAME PACKETS OF
00240 ; 1 TO 7 FRAMES/PACKET ARE SEQUENTIALLY ACCOMODATED.
00250
00260 ; TABLE BEGINNING ON PAGE THREE IS THE LOOK-UP TABLE FOR
00270 ; THE BRILLIANT 'BYTE WISE' CRC SUBROUTINE SUGGESTED BY
00280 ; ARAM PEREZ IN THE THE JUNE '83 ISSUE OF I.E.E.E. MICRO.
00290 ; THE TABLE VALUES FOR THE IBM SDLC 'CRC' WERE GENERATED
00300 ; BY W4UCH AS THE PEREZ PAPER ONLY GAVE THE CRC16 VALUES.
00310
00320 CRCVAL  DEFW   0              ;RECEIVE CRC VALUE STASH
00330 ENDCRC  DEFW   0              ;XMIT CRC VALUE STASH
00340 CRC1    LD     HL,ADDREZ      ;BEGIN MESSAGE LOCATION
00350         LD     BC,(LENG1)     ;LENGTH OF FRAME IN BYTES
00360         LD     DE,65535       ;INTIALIZE DIVIDEND 1'S
00370         CALL   CRCT           ;GENERATE CRC LINE 490
00380         CALL   FINCRC         ;SORT/STUFF RIGHT ORDER
00390         LD     A,(SIGN2)      ;DISPLAY CRC VALUE -
00400         CP     1              ;ON VIDEO DISPLAY ?
00410         RET    Z              ;IF NOT, RETURN.
00420         LD     HL,(ENDCRC)    ;IF SO, THEN DISPLAY IT
00430         CALL   DZ             ;ON TOP LINE OF VIDEO.
00440 CRC2    LD     BC,960         ,= 15 LINES OF VIDEO
00450         LD     HL,ADDREZ      ;BEGIN PACKET ADDRESS
00460         LD     DE,15424       ;2ND LINE OF VIDEO
00470         LDIR                  ;DISPLAY MESSAGE SENT
00480         RET                   ;RETURN WHENCE U CAME +1
00490 CRCT    LD     A,(HL)         ;FIRST BYTE TO CRC
00500         INC    HL             ;INCREMENT FOR NEXT ONE
00510         PUSH   BC             ;SAVE BYTES TO CRC
00520         PUSH   HL             ;SAVE NEXT BYTE LOCATION
00530         XOR    E              ;XOR REMAINDER LSB W/'A'
00540         LD     C,A            ;SAVE RESULT IN 'C'
00550         LD     B,0            ;ZERO OUT 'B'
00570         ADD    HL,BC          ;ADD BC TO LOCATION
00580         ADD    HL,BC          ;ADD BC TO LOCATION
00590         LD     A,D            ;REMAINDER MSB TO 'A'
00600         XOR    (HL)           ;XOR WITH TABLE VALUE
00610         LD     E,A            ;SAVE RESULT IN 'E'
00620         INC    HL             ;NEXT TABLE LOCATION
00630         LD     D,(HL)         ;SAVE VALUE IN 'D'
00640         POP    HL             ;NEXT BYTE TO CRC MEM
00650         POP    BC             ;NUMBER BYTES TO CRC
00660         DEC    BC             ;LESS ONE
00670         LD     A,B            ;TEST FOR
00680         OR     C              ;ZERO
```

```
00690          JP      NZ,CRCT          ;IF NOT, CRC NEXT ONE
00700          RET                      ;ELSE ALL DONE. RETURN
00710 FINCRC   LD      A,E              ;DE = CRC 2 BYTE VALUE
00720          CPL                      ;COMPLEMENT IT
00730          LD      HL,(WHER4B)      ;END OF MESSAGE +1
00740          LD      (HL),A           ;LD 1ST CRC ON MESSAGE
00750          LD      (ENDCRC+1),A     ;AND SAVE IT HERE
00760          INC     HL               ;NEXT MESSAGE LOCATION
00770          LD      A,D              ;SECOND CRC BYTE
00780          CPL                      ;COMPLEMENT IT
00790          LD      (HL),A           ;LD 2ND CRC ON MESSAGE
00800          LD      (ENDCRC),A       ;AND SAVE IT HERE
00810          RET                      ;RETURN WHENCE U CAME +1
00820
00830 ; FOLLOWING IS RECEIVE CRC CHECK FOR EACH FRAME. IT IS
00840 ; CALLED WITH 'BC' REGISTER ALREADY HAVING THE TOTAL
00850 ; NUMBER OF BYTES IN THE FRAME (INCLUDING CRC BYTES).
00860
00870  RCRC    LD      DE,65535         ;RECEIVE CRC CHECK
00880          LD      HL, (BGINIT)     ;BEGIN FRAME LOCATION
00890          CALL    CRCT             ;CRC ALL INCLUDING CRC
00900          LD      (CRCVAL),DE      ;SAVE REMAINDER IN MEM
00910          LD      HL,61624         ;COMPARE REMAINDER WITH
00920          RST     18H              ;61624 DECIMAL
00930          JP      NZ,BADCRC        ;NOT ZERO = BAD ONE
00940          RET                      ; OK, SO RETURN
00950 BADCRC   POP     AF               ;ADJUST STACK
00960          POP     AF               ;FOR 2 CALLS
00970          LD      IY,37692         ;<BAD CRC> MESSAGE
009 80         CALL    SHOWIT           ;DISPLAY ON VIDEO
00990          JP      MODE1A           ;GO AWAIT NEXT PACKET
```

------- **CRC LOOKUP** TABLE ------->

FIGURE 4 CONTINUED

This is the 512 byte CRC lookup table printed out as **256** two byte words to save space. The label TABLE is at location **1.**

```
  1 DEFW     0
  2 DEFW  4489
  3 DEFW  8978
  4 DEFW 12955
  5 DEFW 17956
  6 DEFW 22445
  7 DEFW 25910
  8 DEFW 29887
  9 DEFW 35912
 10 DEFW 40385
 11 DEFW 44890
 12 DEFW 48851
 13 DEFW 51820
 14 DEFW 56293
 15 DEFW 59774
 16 DEFW 63735
 17 DEFW  4225
 18 DEFW   264
 19 DEFW 13203
 20 DEFW  8730
 21 DEFW 22181
 22 DEFW 18220
 23 DEFW 30135
 24 DEFW 25662
 25 DEFW 40137
 26 DEFW 36160
 27 DEFW 49115
 28 DEFW 44626
 29 DEFW 56045
 30 DEFW 52068
 31 DEFW 63999
 32 DEFW 59510
 33 DEFW  8450
 34 DEFW 12427
 35 DEFW   528
 36 DEFW  5017
 37 DEFW 26406
 38 DEFW 30383
 39 DEFW 17460
 40 DEFW 21949
 41 DEFW 44362
 42 DEFW 48323
 43 DEFW 36440
 44 DEFW 40913
 45 DEFW 60270
 46 DEFW 64231
 47 DEFW 51324
 48 DEFW 55797
 49 DEFW 12675
 50 DEFW  8202
 51 DEFW  4753
 52 DEFW   792
 53 DEFW 30631
 54 DEFW 26158
 55 DEFW 21685
 56 DEFW 17724
 57 DEFW 48587
 58 DEFW 44098
 59 DEFW 40665
 60 DEFW 36688
 61 DEFW 64495
 62 DEFW 60006
 63 DEFW 55549
 64 DEFW 51572
 65 DEFW 16900
 66 DEFW 21389
 67 DEFW 24854
 68 DEFW 28831
 69 DEFW  1056
 70 DEFW  5545
 71 DEFW 10034
 72 DEFW 14011
 73 DEFW 52812
 74 DEFW 57285
 75 DEFW 60766
 76 DEFW 64727
 77 DEFW 34920
 78 DEFW 39393
 79 DEFW 43898
 80 DEFW 47859
 81 DEFW 21125
 82 DEFW 17164
 83 DEFW 29079
 84 DEFW 24606
 85 DEFW  5281
 86 DEFW  1320
 87 DEFW 14259
 88 DEFW  9786
 89 DEFW 57037
 90 DEFW 53060
 91 DEFW 64991
 92 DEFW 60502
 93 DEFW 39145
 94 DEFW 35168
 95 DEFW 48123
 96 DEFW 43634
 97 DEFW 25350
 98 DEFW 29327
 99 DEFW 16404
100 DEFW 20893
101 DEFW  9506
102 DCFW 13483
103 DEFW  1584
104 DEFW  6073
105 DEFW 61262
106 DEFW 65223
107 DEFW 52316
108 DEFW 56789
109 DEFW 43370
110 DEFW 47331
111 DEFW 35448
112 DEFW 39921
113 DEFW 29575
114 DEFW 25102
115 DEFW 20629
116 DEFW 16668
117 DEFW 13731
118 DEFW  9258
119 DEFW  5809
120 DEFW  1848
121 DEFW 65487
122 DEFW 60998
123 DEFW 56541
124 DEFW 52564
125 DEFW 47595
126 DEFW 43106
127 DEFW 39673
128 DEFW 35696
129 DEFW 33800
130 DEFW 38273
131 DEFW 42778
132 DEFW 46739
133 DEFW 49708
134 DEFW 54181
135 DEFW 57662
136 DEFW 61623
137 DEFW  2112
138 DEFW  6601
139 DEFW 11090
140 DEFW 15067
141 DEFW 20068
142 DEFW 24557
143 DEFW 28022
144 DEFW 31999
145 DEFW 38025
146 DEFW 34048
147 DEFW 47003
148 DEFW 42514
149 DEFW 53933
150 DEFW 49956
151 DEFW 61887
152 DEFW 57398
153 DEFW  6337
154 DEFW  2376
155 DEFW 15315
156 DEFW 10842
157 DEFW 24293
158 DEFW 20332
159 DEFW 32247
160 DEFW 27774
167 DEFW 42250
162 DEFW 46211
163 DEFW 34328
164 DEFW 38801
165 DEFW 58158
166 DEFW 62119
167 DEFW 49212
168 DEFW 53685
169 DEFW 10562
170 DEFW 14539
171 DEFW  2640
172 DEFW  7129
173 DEFW 28518
174 DEFW 32495
175 DEFW 19572
176 DEFW 24061
177 DEFW 46475
178 DEFW 41986
179 DEFW 38553
180 DEFW 34576
181 DEFW 62383
182 DEFW 57894
183 DEFW 53437
184 DEFW 49460
185 DEFW 14787
186 DEFW 10314
187 DEFW  6865
188 DEFW  2904
189 DEFW 32743
190 DEFW 28270
191 DEFW 23797
192 DEFW 19836
193 DEFW 50700
194 DEFW 55173
195 DEFW 58654
196 DEFW 62615
197 DEW  32808
198 DEFW 37281
199 DEFW 41786
200 DEFW 45747
201 DEFW 19012
202 DEFW 23501
203 DEFW 26966
204 DEFW 30943
205 DEFW  3168
206 DEFW  7657
207 DEFW 12146
208 DEFW 16123
209 DEFW 54925
210 DEFW 50948
211 DEFW 62879
212 DEFW 58390
213 DEFW 37033
214 DEFW 33056
215 DEFM 46011
216 DEFW 41522
217 DEFW 23237
218 DEFW 19276
219 DEFW 31191
220 DEFW 26718
221 DEFW  7393
222 DEFW  3432
223 DEFW 16371
224 DEFW 11898
225 DEFW 59150
226 DEFW 63111
227 DEFW 50204
228 DEFW 54677
229 DEFW 41258
230 DEFW 45219
231 DEFW 33336
232 DEFW 37809
233 DEFW 27462
234 DEFW 31439
235 DEFW 18516
236 DEFW 23035
237 DEFW 11618
238 DEFW 15595
239 DEFW  3696
240 DEFW  8185
241 DEFW 63375
242 DEFW 58886
243 DEFW 54429
244 DEFW 50352
245 DEFW 45483
246 DEFW 40993
247 DEFW 37561
248 DEFW 33584
249 DEFW 31687
250 DEFW 27214
251 DEFW 22741
252 DEFW 18780
253 DEFW 15843
254 DEFW 11370
255 DEFW  7921
256 DEFW  3960
```

```
00100 ;                    FIGURE 5
00110
00120 ; TRANSMIT SUBROUTINE:  SINGLE OR MULTI-FRAME 1200 BAUD
00130
00140 ; PACKET WITH REAL-TIME ZERO INSERTION WHERE APPLICABLE
00150
00160
00170 SENPAK  EXX                     ;SWAP ALTERNATE REGISTERS
00180         LD      HL,(NORMFM)     ;NORMAL FRAME LENGTH
00190         LD      DE,(LASTFM)     ;LAST FRAME LAST PACK LEN
00200         LD      A,(FRMCNT)      ;FRAMES PER PACKET
00210         LD      B,A             ;SAVE IN ALTERNATE 'B'
00220         LD      A,(TESCNT)      ;LAS FRM LAS PACK POINTER
00230         LD      C,A             ;SAVE IN ALTERNATE 'C'
00240         EXX                     ;RESTORE REG. REGISTERS
00250         LD      IY,(STARPK)     ;ASSEMBLED PACK BEGIN ADR
00260         LD      A,1             ;LAST BIT VALUE POINTER
00270         LD      (LASONE),A      ;SAVE IT IN LASONE
00280         ED      (SIGN6),A       ;SET XMIT LO-MEM POINTER
00290         XOR     A               ;ZERO OUT
00300         LD      (ZEROMK),A      ;MARKS IN A ROW COUNTER
00310         LD      (ZEROSP),A      ;SPACES IN A ROW COUNTER
00320 FLGDLY  LD      A,(BK)          ;BACKOFF DELAY 'ON' ?
00330         CP      1               ;IF SO, DO RANDOM
00340         CALL    Z,BAKOFF        ;BACKOFF AFTER CLEAR
00350 FLGNUM  LD      A,60            ;NUMBER FLAGS YOU INPUT
00360 FLG     DEC     A               ;MINUS 1
00370         JP      Z,SEND7         ;IF DONE SEND DATA IN 470
00380         PUSH    AF              ;NUMBER FLAGS REMAINING
00390         CALL    FLAG            ;SEND SDLC/HDLC FLAG
00400         POP     AF              ;RESTORE FLAG COUNTER
00410         JP      FLG             ;DO NEXT ONE
00420 FLAG    LD      HL,98           ;1200 BAUD COUNT NUMBER
00430         LD      (SPEED),HL      ;STASH IT IN SPEED
00440         LD      A,126           ;FLAG BYTE VALUE
00450         CALL    SN1A            ;NO ZERO INSERT TRANSMIT
00460         RET                     ;RETURN TO LINE 400
00470 SEND7   EXX                     ;SWAP ALTERNATE REGISTERS
00480         PUSH    HL              ;FRAME LENGTH TO STACK
00490         EXX                     ;RESTORE REG. REGISTERS
00500         POP     DE              ;FRAME LENGTH TO 'DE'
00510         DEC     DE              ;DECREMENT FRAME LENGTH
00520         LD      A,D             ;TEST
00530         OR      E               ;FOR ZERO
00540         JP      Z,KYBD4         ;IF ZERO, GOTO LINE 600
00550         PUSH    DE              ;FRAME LENGTH ON STACK
00560         LD      A,(IY)          ;BYTE TO TRANSMIT
00570         INC     IY              ;NEXT BYTE LOCATION
00580         CALL    SN1             ;ZERO INSERTION TRANSMIT
00590         JP      SEND7+3         ;GOTO LINE 500
00600 KYBD4   CALL    FLAG            ;XMIT FRAME CLOSING FLAG
00610         EXX                     ; SWAP ALTERNATE REGISTERS
00620         DEC     C               ;LAST FRAME LAST PACK ?
00630         JP      Z,KYBD4A        ;IF ZERO JUMP TO LINE 670
00640         DEC     B               ; DEC NORMAL FRAMES/PACK
00650         JP      NZ,SEND7+1      ;NOT ZERO, GOTO LINE 480
00660         JP      DUN1            ;IF ZERO, GOTO LINE 690
00670 KYBD4A  PUSH    DE              ;LAST FRAME LAST PACK LEN
00680         JP      SEND7+2         ;GOTO LINE 490
00690 DUN1    XOR     A               ;SWITCH T/R RELAY
00700         OUT     (0),A           ;TO RECEIVE
00710         EXX                     ;RESTORE REG. REGISTERS
00720         JP      MODE1           ;GOTO RECEIVE MODE
00730 SN1     LD      D,A             ;BYTE VALUE TO TRANSMIT
00740         LD      E,8             ;NUMBER OF BITS PER BYTE
00750 SN2     LD      A,(LASONE)      ;1 = SPACE & 5 = MARK
00760         CP      1               ;WAS IT A SPACE ?
00770         JP      Z,LASTSP        ;IF SO, GOTO LAST SPACE
00780         BIT     0,D             ;SET Z FLAG FOR BIT ZERO
00790         CALL    NZ,MARK         ;IF NOT ZERO SEND MARK
00800         BIT     0,D             ;SET Z FLAG FOR BIT ZERO
00810         CALL    Z,SPACE         ;IF ZERO SEND SPACE
00820         NOP                     ;2 USEC TIMING ADJUST
00830         DEC     E               ;-1 FROM BIT COUNTER
00840         RET     Z               ;IF ZERO, RETURN LINE 590
00850         RRC     D               ;RIGHT SHIFT ALL 1 BIT
00860         JP      SN2             ;GO BACK FOR NEXT BIT
00870 LASTSP  BIT     0,D             ;SET Z FLAG FOR BIT ZERO
00880         CALL    NZ,SPACE        ;IF NOT ZERO SEND SPACE
00890         BIT     0,D             ;SET Z FLAG FOR BIT ZERO
00900         CALL    Z,MARK          ;IF ZERO SEND MARK
00910         NOP                     ;2 USEC TIMING ADJUST
00920         DEC     E               ;-1 FROM BIT COUNTER
00930         RET     Z               ;IF ZERO, RETURN LINE 590
00940         RRC     D               ;RIGHT SHIFT ALL 1 BIT
00950         JP      SN2             ;GO BACK FOR NEXT BIT
00960 SPACE   LD      A,5             ;SEND SPACE TONE
00970         OUT     (0),A           ;VIA PORT ZERO
00980         XOR     A               ;ZERO OUT 'A' REGISTER
00990         LD      (ZEROMK),A      ;AND ZERO MARK COUNTER
01000         LD      A,(SPEED)       ;COUNTDOWN VALUE
01010         LB      HL,SPACEA       ;RETURN MEM LOCATION
01020         PUSH    HL              ;PUSH ON TOP OF STACK
01030         LD      HL,DECSP        ;JP (HL) ADDRESS
01040 DECSP   DEC     A               ;-1 COUNTDOWN VALUE
01050         RET     Z               ;GOTO SPACEA WHEN ZERO
01060         JP      (HL)            ;JUMP TO DECSP
01070 SPACEA  LD      A,(LASONE)      ;PREVIOUS BIT SENT
01080         CP      5               ;WAS IT A MARK ?
01090         JP      Z,SPACEB        ;IF SO, DON'T COUNT IT
01100         LD      A,(ZEROSP)      ;SPACE COUNTER STASH
01110         INC     A               ;+1 TO SPACE COUNTER
01120         CP      5               ;5 SPACES IN A ROW ?
01130         JP      Z,SPACEC        ;IF SO, DO ZERO INSERTION
01140         LD      (ZEROSP),A      ; IF NOT, SAVE NEW VALUE
01150         NOP                     ;2 USEC TIMING ADJUST
01160         RET                     ;RETURN WHENCE U CAME +1
01170 SPACEB  LD      A,1             ;SINCE NOT SAME CHANGE IT
01180         LD      (LASONE),A      ;UPDATE LASTONE
01190         NOP                     ;EQUALIZING DELAY
01200         NOP                     ;EQUALIZING DELAY
01210         NOP                     ;EQUALIZING DELAY
01220         RET                     ;RETURN WHENCE U CAME +1
01230 SPACEC  LD      A,1             ;1 = SPACE & 5 = MARK
01240         LD      (LASONE),A      ;UPDATE LASTONE
01250         LD      BC,1            ;DELAY - NO SN2 ITERATION
01260         CALL    060H            ;APPROX. 30 MICROSECONDS
01270         CALL    MARK            ;DO ZERO INSERTION
```

```
01280        XOR    A              ;ZERO OUT 'A' REGISTER
01290        LD     (ZEROMK),A     ;AND ZERO MARK COUNTER
01300        RET                   ;RETURN WHENCE U CAME +1
01310 SPACE1 LD     A,5            ;1310-1410 ONLY FOR FLAG
01320        OUT    (0),A          ;SPACE TONE PORT ZERO
01330        LD     A,1            ;1 = SPACE & 5 = MARK
01340        LD     (LASONE),A     ;UPDATE LASTONE
01350        XOR    A              ;ZERO OUT 'A' REGISTER
01360        LD     (ZEROMK),A     ;AND ZERO MARK COUNTER
01370        LD     A,(SPEED)      ;COUNTDOWN VALUE
01380        LD     HL,DECSP1      ;JP (HL) ADDRESS
01390 DECSP1 DEC    A              ;-1 COUNTDOWN VALUE
01400        RET    Z              ;RETURN WHENCE U CAME +1
01410        JP     (HL)           ;JUMP TO DECSP1
01420 MARK   LD     A,1            ;SEND MARK TONE
01430        OUT    (0),A          ;VIA PORT ZERO
01440        XOR    A              ;ZERO OUT 'A' REGISTER
01450        LD     (ZEROSP),A     ;AND ZERO SPACE COUNTER
01460        LD     A,(SPEED)      ;COUNTDOWN VALUE
01470        LD     HL,MARKA       ;RETURN MEM LOCATION
01480        PUSH   HL             ;PUSH ON TOP OF STACK
01490        LD     HL,DECMK       ;JP (HL) ADDRESS
01500 DECMK  DEC    A              ;-1 COUNTDOWN VALUE
01510        RET    Z              ;GOTO MARKA WHEN ZERO
01520        JP     (HL)           ;JUMP TO DECMK
01530 MARKA  LD     A,(LASONE)     ;PREVIOUS BIT SENT
01540        CP     1              ;WAS IT A SPACE ?
01550        JP     Z,MARKB        ;IF SO, DON'T COUNT IT
01560        LD     A,(ZEROMK)     ;MARK COUNTER STASH
01570        INC    A              ;+1 TO MARK COUNTER
01580        CP     5              ;5 MARKS IN A ROW ?
01590        JP     Z,MARKC        ;IF SO, DO ZERO INSERTION
01600        LD     (ZEROMK),A     ;IF NOT, SAVE NEW VALUE
01610        NOP                   ;2 USEC TIMING ADJUST
01620        RET                   ;RETURN WHENCE U CAME +1
01630 MARKB  LD     A,5            ;SINCE NOT SAME CHANGE IT
01640        LD     (LASONE),A     ;UPDATE LASTONE
01650        NOP                   ;EQUALIZING DELAY
01660        NOP                   ;EQUALIZING DELAY
01670        NOP                   ;EQUALIZING DELAY
01680        RET                   ;RETURN WHENCE U CAME +1
01690 MARKC  LD     A,5            ;1 = SPACE & 5 = MARK
01700        LD     (LASONE),A     ;UPDATE LASTONE
01710        LD     BC,1           ;DELAY - NO SN2 ITERATION
01720        CALL   060H           ;APPROX. 30 MICROSECONDS
01730        CALL   SPACE          ;DO ZERO INSERTION
01740        XOR    A              ;ZERO OUT 'A' REGISTER
01750        LD     (ZEROSP),A     ;AND ZERO SPACE COUNTER
01760        RET                   ;RETURN WHENCE U CAME +1
01770 MARK1  LD     A,1            ;1770-1870 ONLY FOR FLAG
01780        OUT    (0),A          ;SEND MARK TONE
01790        LD     A,5            ;1 = SPACE & 5 = MARK
01800        LD     (LASONE),A     ;UPDATE L&STONE
01810        XOR    A              ;ZERO OUT 'A' REGISTER
01820        LD     (ZEROSP),A     ;AND ZERO SPACE COUNTER
01830        LD     A,(SPEED)      ;COUNTDOWN VALUE
01840        LD     HL,DECMK1      ;JP (HL) ADDRESS
01850 DECMK1 DEC    A              ;-1 COUNTDOWN VALUE
01860        RET    Z              ;RETURN WHENCE U CAME +1
01870        JP     (HL)           ;JUMP TO DECMK1
01880 SN1A   LD     D,A            ;1880-2080 ONLY FOR FLAG
01890        LD     E,8            ;NUMBER OF BITS PER BYTE
01900 SN2A   LD     A,(LASONE)     ;1 = SPACE & 5 = MARK
01910        CP     1              ;WAS XT A SPACE ?
01920        JP     Z,LASSP        ;IF SO, GOT0 LAST SPACE
01930        BIT    0,D            ;SET Z FLAG FOR BIT ZERO
01940        CALL   NZ,MARK1       ;IF NOT ZERO SEND MARK
01950        BIT    0,D            ;SET Z FLAG FOR BIT ZERO
01960        CALL   Z,SPACE1       ;IF ZERO SEND SPACE
01970        DEC    E              ;-1 FROM BIT COUNTER
01980        RET    Z              ;IF ZERO, RETURN LINE 460
01990        RRC    D              ;RIGHT SHIFT ALL 1 BIT
02000        JP     SN2A           ;GO BACK FOR NEXT BIT
02010 LASSP  BIT    0,D            ;SET Z FLAG FOR BIT ZERO
02020        CALL   NZ,SPACE1      ;IF NOT ZERO SEND SPACE
02030        BIT    0,D            ;SET Z FLAG FOR BIT ZERO
02040        CALL   Z,MARK1        ;IF ZERO SEND MARK
02050        DEC    E              ;-1 FROM BIT COUNTER
02060        RET    Z              ;IF ZERO, RETURN LINE 460
02070        RRC    D              ;RIGHT SHIFT ALL 1 BIT
02080        JP     SN2A           ;GO BACK FOR NEXT BIT
02090 ZEROSP DEFB   0              ;SPACE COUNTER STASH
02100 ZEROMK DEFB   0              ;MARK COUNTER STASH
02110 SPEED  DEFW   98             ;XMIT COUNTDOWN VALUE
02120 LASONE DEFB   1              ;LAST BIT SENT VALUE
02130 ; - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
02140 ;END OF SINGLE/MULTI-FRAME 1200 BAUD SYNCHRONOUS TRANSMIT
02150
02160 ; FOR MODEL III CLOCK CHANGE LINE 420 FROM 98 TO 115.
```

```
00100 ;                        FIGURE 6
00110
00120 ; IN-PROGRAM DISK I/O SUBROUTINES FOR AX.25 PROTOCOL
00130
00140 ; FOR TRSDOS 1.3 - TRSDOS 2.3 - NEWDOS + AND 1.0
00150
12790         ORG     49632           ;SUBROUTINE MEM LOCATION
12800 FCB     DEFS    32              ;DISK FILE CONTROL BLOCK
12810 BUFFER  DEFS    256             ;DISK I/O WORKING SPACE
12820 DIZ     LD      A,(HL)          ;DISPLAY MESSAGE ON VIDEO
12830         CP      0               ;END OF MESSAGE DELIMITER
12840         JP      Z,FINISH        ; IF ZERO, ALL DONE
12850         CALL    033H            ;DISPLAY BYTE ON VIDEO
12860         INC     HL              ;NEXT MESG BYTE LOCATION
12870         JP      DIZ             ;GO DISPLAY NEXT BYTE
12880 FINISH  RET                     ;RETURN WHENCE U CAME +i
12890 INPNAM  CALL    CLS             ;C L E A R   V I D E O
12900         LD      HL,NAM1         ;REMEMBER DELIMITERS MSG?
12910         CALL    DIZ             ;DISPLAY XT ON VIDEO
12920         CALL    049H            ;AWAIT KEYBOARD INPUT
12930         CP      1               ;BREAK KEY PRESSED ?
12940         JP      Z,ESCAPE        ;IF SO, ESCAPE LINE 13180
12950         CALL    CLS             ;C L E AR   V I D E O
12960         LD      HL,NAM1A        ;INPUT BEGIN ADDRESS MSG?
12970         CALL    DIZ             ;DISPLAY IT ON VIDEO
12980         CALL    1BB3H           ;KEYBOARD INPUT ROUTINE
12990         RST     10H             ;SCAN STRING SET 'C' FLAG
13000         CALL    1E5AH           ;CONVERT UNSIGNED INTEGER
13010         EX      DE,HL           ;PUT INTEGER IN HL
13020         LD      (DUMP+1),HL     ;STUFF BEGIN ADDRESS DUMP
13030         LD      (HOWFAR+1),HL   ;AND IN HOWFAR MEM
13040 INNAME  CALL    CLS             ;C L E AR   V I D E O
13050         LD      HL,NAM2         ;INPUT FILE NAME MESSAGE?
13060         CALL    DIZ             ;DISPLAY IT ON VIDEO
13070         CALL    1BB3H           ;KEYBOARD INPUT ROUTINE
13080         LD      HL,41E8H        ;WHERE STASHED IN MEM
13090         LD      A,(HL)          ;FIRST BYTE OF FILE NAME
13100         CP      0               ;YOU INPUT NOTHING ?
13110         JP      Z,ESCAPE        ;IF SO, ESCAPE LINE 13180
13120         CALL    LONG            ;HOW MANY BYTES IN NAME ?
13130         LD      HL,41E8H        ;NAME ADDRESS IN MEM
13140         LD      DE,FCB          ;FILE CONTROL BLOCK ADR
13150         LDIR                    ;MOVE TO CONTROL BLOCK
13160         CALL    DRIVE           ;AND MOVE DRIVE NO. TOO
13170         RET                     ;RETURN WHENCE U CAME +1
13180 ESCAPE  POP     AF              ;ADJUST STACK FOR CALL
13190         LD      HL,53248        ;RESET TO NORMAL
13200         LD      (DUMP+1),HL     ;DUMP AND
13210         LD      (HOWFAR+1),HL   ;HOWFAR
13220         JP      MENU            ;TIP MENU FOR INSTRUCTS
13230 LONG    LD      BC,0            ;HOW LONG IS FILE NAME ?
13240 LON1    LD      A,(HL)          ;BYTE FROM NAME STRING
13250         CP      0               ;ZERO DELIMITER
13260         RET     Z               ;RETURN WITH COUNT
13270         INC     C               ;1 MORE BYTE
13280         INC     HL              ;NEXT MEM LOCATION
13290         JP      LON1            ;GO COUNT IT
13300 LBYTES  DEFW    0               ;NUMBER BYTES READ STASH
13310 CLRLO   LD      HL,16872        ;CLEAR
13320         LD      DE,16873        ;LOW MEMORY
13330         LD      BC,12878        ;WITH
13340         LD      (HL),0          ;ZEROS
13350         LDIR                    ;DO IT RIGHT NOW
13360         RET                     ;RETURN WHENCE U CAME +1
13370 OPEN1   LD      DE,FCB          ;FILE CTRL BLOCK MEM ADR
13380         LD      HL,BUFFER       ;DISK I/O BUFFER ADDRESS
13390         LD      B,0             ;256 BYTE RECORD LENGTH
13400         CALL    4424H           ;OPEN AN EXISTING FILE
13410         JR      NZ,ERROR        ;Z FLAG SET IF ERROR
13420         RET                     ;RETURN WHENCE U CAME +1
13430 READ    LD      HL,53248        ;WHERE TO PUT FILE IN MEM
13440         LD      DE,FCB          ;FILE CTRL BLOCK ADDRESS
13450 LG      PUSH    HL              ;SAVE MEM LOCATION STACK
13460         CALL    13H             ;READ BYTE FROM DISK FILE
13470         POP     HL              ;RESTORE MEM LOCATION
13480         LD      (HL),A          ;AND LOAD IT IN MEM
13490         INC     HL              ;NEXT MEM LOCATION
13500         PUSH    HL              ;SAVE IT IN STACK
13510         PUSH    DE              ;SAVE FCB POINTER
13520 LONG1   LD      DE,65535        ;FILE END ADDRESS IN MEM
13530         OR      A               ;CLEAR CARRY FLAG
13540         SBC     HL,DE           ;SUB HL - DE SET Z FLAG
13550         POP     DE              ;RESTORE FCB POINTER
13560         POP     HL              ;RESTORE MEM LOCATION
13570         RET     Z               ;RETURN IF ALL DONE
13580         JP      LG              ;GO READ NEXT BYTE
13590 CLOSE   LD      DE,FCB          ;FILE CTRL BLOCK ADDRESS
13600         CALL    4428H           ;CLOSE FILE SUBROUTINE
13610         PUSH    AF              ;SAVE IN STACK
13620         LD      HL,53248        ;BEGIN HI-MEM ADDRESS
13630         LD      (DUMP+1),HL     ;RESET DUMP
13640         LD      (HOWFAR+1),HL   ;RESET HOWFAR
13650         POP     AF              ;RESTORE AF
13660         RET     Z               ;RETURN UNLESS ERROR
13670         POP     HL              ;ADJUST STACK FOR CALL
13680 ERROR   LD      H,0             ;ZERO OUT 'H'
13690         LD      L,A             ;ERROR NUMBER TO 'L'
13700         CALL    0A9AH           ;MOVE HL INTO ACCUM
13710         CALL    0A7FH           ;MAKE SURE AN INTEGER
13720         CALL    OFBDH           ;CONVERT TO STRING
13730         LD      DE,MS2C+9       ;ERROR MESSAGE LOCATION
13740 ER1     LD      A,(HL)          ;ERROR NUMBER
13750         CP      0               ;ZERO STRING DELIMITER
13760         JP      Z,ER2           ;ALL DONE ? GOTO ER2
13770         LD      (DE),A          ;ERROR NUMBER TO MEM
13780         INC     HL              ;NEXT ERROR # LOCATION
13790         INC     DE              ;NEST MESSAGE LOCATION
13800         JP      ER1             ;GO MOVE NEXT ONE
13810 ER2     CALL    CLS             ;C L E A R   V I D E O
13820         POP     AF              ;ADJUST STACK
13830         CALL    SETUP           ;RESTORE PGM POINTERS
13840         CALL    CLRLO           ;CLEAR OUT DOS
13850         CALL    CLRHY           ;CLEAR OUT HI-MEM
13860         LD      HL,MS2C         ;ERROR # MESSAGE
13870         CALL    DIZ             ;DISPLAY IT ON VIDEO
13880         CALL    049H            ;PRESS ANY KEY
13890         JP      MENU            ;GOTO MENU FOR INSTRUCTS
13900 DRIVE   LD      A,@:'           ;DRIVE # SEPARATOR
```

```
13910        LD      (DE),A         ;FILE CONTROL BLOCK
13920        LD      (BC),A         ;FUTURE USE VOL. 3
13930        INC     DE             ;NEXT FCB LOCATION
13940        INC     Bc             ;FUTURE USE VOL. 3
13950        LD      A,'1'          ;DRIVE# CHANGE UR CHOICE
13960        LD      (DE),A         ;INTO FILE CTRL BLOCK
13970        LD      (BC),A         ;FUTURE USE VOL. 3
13980        INC     DE             ;FCB NEXT LOCATION
13990        INC     Bc             ;FUTURE USE VOL. 3
14000        LD      A,13           ;FCB DELIMITER
14010        LD      (DE),A         ;INTO FILE CTRL BLOCK
14020        LD      (BC),A         ;FUTURE USE VOL. 3
14030        RET                    ;RETURN WHENCE U CAME +i
14040 NAM2   DEFM    'INPUT FILE NAME '
14050        DEFB    0              ;DELIMITER
14060 OPEN3  LD      HL,BUFFER      ;DISK I/O BUFFER ADDRESS
14070        LD      DE,FCB         ;FILE CTRL BLOCK ADDRESS
14080        LD      B,0            ;256 BYTES PER RECORD
14090        LD      C,10H          ;FILE TYPE DOUBTFUL
14100        CALL    4420H          ;OPEN NEW DISK FILE
14110        RET                    ;RETURN WHENCE U CAPE +i
14120 HOWFAR LD      HL,53248       ;CALCULATE BYTES TO SAVE
14130 FAR1   INC     HL             ;TO DISK FILE
14140        LD      A,(HL)         ;LOOK
14150        CP      128            ;FOR
14160        JP      NZ,FAR1        ;THREE
14170        INC     HL             ;EACH
14180        LD      A,(HL)         ;DECIMAL
14190        CP      128            ;128
14200        JP      NZ,FARi        ;END
14210        INC     HL             ;OF
14220        LD      A,(HL)         ;MESSAGE
14230        CP      128            ;DELIMITERS
14240        JP      NZ,FARi        ;IN A
14250        INC     HL             ;ROW
14260        LD      (SOFAR+i),HL   ;SAVE THEM IN SOFAR
14270        RET                    ;RETURN WHENCE U CAME +1
14280 DUMP   LD      HL,53248       ;BEGIN DATA LOCATION
14290        LD      DE,FCB         ;FILE CTRL BLOCK ADDRESS
14300 DUM1   LD      A,(HL)         ;BYTE TO SAVE ON DISK
14310        PUSH    HL             ;SAVE BYTE MEM LOCATION
14320        CALL    1BH            ;WRITE TO DISK SUBROUTINE
14330        POP     HL             ;RESTORE BYTE LOCATION
14340        JP      NZ,ERROR       ;Z FLAG SET IF ERROR
14350        INC     HL             ;NEXT BYTE LOCATION
14360        PUSH    HL             ;SAVE IT IN STACK
14370        PUSH    DE             ;SAVE FCB POINTER
'I4380 SOFAR LD      DE,65535       ;LAST MEM BYTE LOCATION
14390        OR      A              ;CLEAR CARRY FLAG
14400        SBC     HL,DE          ;SUBTRACT HL MINUS DE
14410        POP     DE             ;RESTORE FCB POINTER
14420        POP     HL             ;AND NEXT MEM LOCATION
14430        RET     Z              ;RETURN IF ALL DONE
14440        JP      DUM1           ;GO DUMP NEXT ONE TO DISK
14450 NAM1   DEFM    'REMEMBER 128 DELIMITERS ? HIT BREAK TO ES
CAPE ELSE <ENTER>'
14460        DEFB    0              ;DELIMITER
14470        ORG     0C840H         ;SAVE FILE MEM LOCATION
14480 SVFILE CALL    INPNAM         ;INPUT FILE NAME
14490        CALL    HOWFAR         ;CALCULATE BYTES TO SAVE
14500        LD      A,195          ;RESTORE JUMP
14510        LD      (400CH),A      ;TO LOW MEMORY
14520        CALL    MOVDN          ;MOVE DOS BACK DOWN MEM
14530        CALL    OPEN3          ;OPEN OR CREATE DISK FILE
14540        CALL    DUMP           ;DUMP IT TO DISK
14550        CALL    'CLOSE         ;CLOSE TME DISK FILE
14560        LD      SP,29758       ;RESET STACK POINTER
14570        CALL    SETUP          ;REINITIALIZE PGM PTRS
14580        CALL    CLRLO          ;CLEAR OUT DOS LO-MEM
14590        JP      MENU           ;GO FOR INSTRUCTIONS
14600 MS2C   DEFM    'ERROR #       ;DISK I/O ERROR MESSAGE
14610        DEFB    0              ;DELIMITER
14620        ORG     0C880H         ;LOAD FILE MEM LOCATION
14630 LDFILE CALL    INNAME         ;INPUT FILE NAME
14640        CALL    CLRHY          ;CLEAR HI-MEMORY
14650        LD      A,195          ;RESTORE JUMP
14660        LD      (400CH),A      ;TO LOW MEMORY
14670        CALL    MOVDN          ;MOVE DOS BACK DOWN MEL;
14680        CALL    OPEN1          ;OPEN AN EXISTING FILE
14690        CALL    MULPLY         ;CALCULATE FILE LENGTH
14700        CALL    READ           ;LOAD FILE TO HI-MEMORY
14710        LD      (HIHL),HL      ;SAVE HI-MEM END OF FILE
14720        CALL    CLOSE          ;CLOSE DISK FILE
14730        LD      SP,29758       ;RESET STACK POINTER
14740        CALL    SETUP          ;REINITIALIZE PGM PTRS
14750        CALL    CLRLO          ;CLEAR OUT DOS LO-MEM
14760        CALL    BAKUP          ;CHECK EXTRA FILE DATA ?
14770        JP      MENU           ;MENU FOR INSTRUCTIONS
14780 HIHL   DEFW    0              ;END HI-MEM FILE STASH
14790 BAKUP  LD      HL,(HIHL)      ;! ! VOLUME 3 ONLY ! !
14800 BAK1   DEC     HL             ;THIS
14810        LD      A,(HL)         ;SCINTILLATING
14820        CP      28             ;ROUTINE
14830        JP      Z,BAK1         ;TRIES
14840        CP      128            ;TO
14850        JP      Z,TESAGN       ;BACKUP
14860        INC     HL             ;IN HI-MEMORY
14870        LD      (BEFORE),HL    ;TILL IT
14880        RET                    ;IT
14890 TESAGN DEC     HL             ;FINDS
14900        LD      A,(HL)         ;3 EACH
14910        CP      128            ;128'S IN
14920        JP      NZ,BAK1        ;A ROW.
14930        DEC     HL             ;WHEN FOUND
14940        LD      A,(HL)         ;IT RESETS BEFOR.
14950        CP      128            ;SO THAT WHEN THE PROGRAM
14960        JP      NZ,BAK1        ;AUTOMATICALLY MOVES THE
14970        LD      (BEFORE),HL    ;FILE DOWN, ONLY DATA IS
14980        RET                    ;MOVED. ! VOLUME 3 ONLY !
14990 NAM1A  DEFM    'INPUT BEGINNING MEM ADDRESS (53248 NOMINA
L) '
15000        DEFB    0              ;DELIMTER
15010 MULPLY LD      A,(FCB+12)     ;NUMBER RECORDS IN FILE
15020 MUL0   LD      HL,0           ;ZERO OUT BYTE COUNTER
15030 MUL1   LD      DE,256         ;BYTES PER RECORD
15040        ADD     HL,DE          ;ADD THEM UP
15050        DEC     A              ;MINUS ONE RECORD
15060        JP      Z,MUL2         ;ALL DONE, GOTO MUL2
```

```
15070          JP     MUL1            ;ADD UP NEXT ONE
15080  MUL2    LD     A,(FCB+8)       ;BYTES IN LAST SECTOR
15090          LD     E,A             ;STUFF IN 'E'
15100          LD     D,0             ;ZERO OUT 'D'
15110          ADD    HL,DE           ;ADD THEM UP
15120  MUL3    LD     (LBYTES),HL     ;AND SAVE THEM HERE
15130          LD     DE,53248        ;BEGIN HIGH MEMORY
15140          ADD    HL,DE           ;ADD BYTES TO HI-MEM
15150          LD     (LONG1+1),HL    ;AND SAVE THEM HERE
15160          RET                    ;RETURN WHENCE U CAME +1
15170
15180 ; - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
15190 ; END OF VOLUME 2 - DISK I/O SUBROUTINES
```

```
00100 ;                          FIGURE 7
00110
00120 ; IN-PROGRAM EDIT/MODIFY/MONITOR SUBROUTINE - 866 BYTES
00130
00140 ; ALSO USED FOR KEYBOARD INPUT PACKET MESSAGES
00150
00160
05230          ORG    38912           ;SUBROUTINE MEM LOCATION
05240 DISMEM   LD     HL,40960        ;CURRENT PACK LOCATION
05250 DISEM1   LD     (MEMO),HL       ;TOP OF PAGE STASH
05260 DISPLA   LD     HL,(MEMO)       ;BACK TO HL REGISTER
05270          LD     (LASMEM),HL     ;INC/DEC STASH
05280          DEC    HL              ;MINUS ONE
05290          LD     (MEMO1),HL      ;BOTTOM PREVIOUS PAGE
05300          INC    HL              ;TOP OF THIS PAGE OF MEM
05310          LD     DE,15360        ;BEGINNING VIDEO MEMORY
05320          LD     BC,1024         ;BYTES PER PAGE OF VIDEO
05330 AGAIN    LD     A,(HL)          ;CHANGE MODEL III
05340          BIT    7,A             ;VIDEO DISPLAY
05350          CALL   Z,SET6          ;TO SIMILAR TO
05360          BIT    7,A             ;MODEL I
05370          CALL   NZ,RES6         ;VIDEO DISPLAY
05380          LD     (DE),A          ;STASH BYTE IN VIDEO
05390          INC    HL              ;NEXT BYTE FROM MEMORY
05400          INC    DE              ;NEXT VIDEO DISPLAY MEM
05410          DEC    BC              ;BYTES TO MOVE COUNTER
05420          LD     A,B             ;TEST B
05430          CP     0               ;IF ZERO
05440          JP     Z,TESTIT        ;TEST C
05450          JP     AGAIN           ;ELSE MOVE NEXT BYTE
05460 RES6     RES    6,A             ;ZERO OUT BIT 6
05470          RET                    ;RETURN WHENCE U CAME +1
05480 SET6     BIT    6,A             ;TEST BIT 6
05490          RET    NZ              ;RETURN IF SET TO 1
05500          BIT    5,A             ;TEST BIT 5
05510          RET    NZ              ;RETURN IF SET TO 1
05520          SET    6,A             ;IF NOT, SET BIT 6 TO 1
05530          RET                    ;RETURN WHENCE U CAME +1
05540 TESTIT   LD     A,C             ;BYTES TO MOVE COUNTER
05550          CP     0               ;ZERO ?
05560          JP     NZ,AGAIN        ;IF NOT, MOVE NEXT ONE
05570          LD     (MEMO),HL       ;TOP NEXT PAGE MEMORY
05580 NEXT     CALL   049H            ;AWAIT KEYBOARD INPUT
05590          CP     1               ;BREAK KEY ?
05600          JP     Z,7630H         ;IF SO, GOTO FTP MENU
05610          CP     13              ;ENTER KEY ?
05620          JP     Z,DISPLA        ;IF SO, DISPLAY NEXT PAGE
05630          CP     45              ;MINUS KEY ?
05640          JP     Z,BACKUP        ;IF SO DISPLAY LOWER PAGE
05650          CP     77              ;'M' KEY PRESSED ?
05660          JP     Z,MODIF         ;IF SO, GOTO MODIFY MODE
05670          JP     NEXT            ;ELSE IGNORE IT
05680 BACKUP   LD     HL,(MEMO1)      ;MOVE THE
05690          INC    HL              ;VIDEO DISPLAY
05700          LD     (MEMO),HL       ;DOWN A FULL PAGE
05710          DEC    HL              ;IN MEMORY
05720          LD     DE,16383        ;LAST BYTE VIDFO MEMORY
05730          LD     BC,1024         ;FULL PAGE VIDEO BYTES
05740 AGAIN1   LD     A,(HL)          ;CHANGE MODEL III
```

```
05750         BIT     7,A           ;VIDEO DISPLAY
05760         CALL    Z,SET6        ;TO SIMILAR TO
05770         BIT     7,A           ;MODEL I
05780         CALL    NZ,RES6       ;VIDEO DISPLAY
05790         LD      (DE),A        ;STASH BYTE IN VIDEO
05800         DEC     HL            ;NEXT LOWER BYTE MEMORY
05810         DEC     DE            ;NEXT LOWER BYTE VIDEO
05820         DEC     BC            ;DECREMENT BYTE COUNTER
05830         LD      A,B           ;TEST B
05840         CP      0             ;IF ZERO
05850         JP      Z,TESIT       ;TEST C
05860         JP      AGAIN1        ;ELSE MOVE NEXT BYTE
05870 TESIT   LD      A,C           ;TEST C
05880         CP      0             ;FOR ZERO
05890         JP      NZ,AGAIN1     ;IF NOT, MOVE NEXT BYTE
05900         LD      (MEMO1),HL    ;BOTTOM NEXT PAGE DOWN
05910         INC     HL            ;TOP THIS PAGE OF MEM
05920         LD      (LASMEM),HL   ;AND SAVE THIS LOCATION
05930         JP      NEXT          ;GO AWAIT NEXT COMMAND
05940 LASMEM  DEFW    0             ;MEM STASH
05950 MODIF   LD      IX,15360      ;MODIFY MODE - MODIFY
05960         LD      IY,(LASMEM)   ;BOTH VIDEO & REAL MEMORY
05970 CONT3   CALL    BLINK-9       ;BLINKING CURSOR
05980         LD      A,(LNFEED)    ;LINEFEED AFTER CARRET?
05990         CP      1             ;IF SO
06000         JP      Z,LFEED2      ;STUFF IT IN MEMORY
06010         CALL    BLINKB        ;RESTORE MEM CHARACTER
06020         LD      A,(14400)     ;KEYBOARD ROW PSUEDO MEM
06030         CP      4             ;BREAK KEY PRESSED ?
06040         JP      Z,NEXT2       ;IF SO, RESUME EDIT MODE
06050         CP      32            ;LEFT ARROW KEY PRESSED ?
06060         JP      Z,LEFT1       ;MOVE CURSOR BACK A SPACE
06070         CP      64            ;RIGHT ARROW KEY PRESSED?
06080         JP      Z,RIGHT1      ;MOVE CURSOR AHEAD SPACE
06090         CP      16            ;DOWN ARROW KEY PRESSED 3
06100         JP      Z,DOWN1       ;MOVE CURSOR DOWN 1 LINE
06110         CP      8             ;UP ARROW KEY PRESSED ?
06120         JP      Z,UPONE       ;MOVE CURSOR UP 1 LINE
06130         LD      A,(14464)     ;SHIFT KEY PSUEDO MEM
06140         CP      0             ;EITHER SHIFTKEY PRESSED?
06150         JP      NZ,NOTASC     ;IF SO, TEST NOT ASCII
06160 CONT3B  CALL    02BH          ;KEYBOARD TO 'A'
06170         CP      11            ;SUBTRACT 11
06180         JP      M,CONT3       ;IF MINUS, IGNORE IT
06190         CP      13            ;ENTER KEY ?
06200         CALL    Z,LFEED1      ;SETUP AUTO LINE FEED
06210         CP      32            ;SPACE ?
06220         JP      Z,CK          ;TEST ILLEGAL SHIFT
06230         CP      64            ;@ KEY 3
06240         JP      Z,CONT3       ;IF SO, IGNORE IT
06250         CP      91            ;UP ARROW ?
06260         JP      Z,CONT3       ;IF SO, IGNORE IT
06270         CP      96            ;SHIFT @ ?
06280         JP      Z,CONT3       ;IF SO, IGNORE IT
06290         LD      (HOLDZ),A     ;SAVE BYTE INPUT
06300         LD      A,(UPSIDE)    ;TEST FOR LOWERCASE
06310         CP      0             ;IF SO
06320         JP      NZ,INVERT     ;INVERT IT
06330         LD      A,(HOLDZ)     ;RESTORE BYTE INPUT
06340         CP      65            ;SUBTRACT 65
06350         JP      M,CONT5       ;MINUS JUMP AROUND RESET
06360         RES     5,A           ;RESET BIT 5
06370 CONT5   LD      (IX),A        ;DISPLAY BYTE ON VIDEO
06380         LD      (IY),A        ;LOAD BYTE INTO RAM MEM
06390         CALL    CKAHED        ;NEXT LOCATION IN BOUNDS?
06400         INC     IX            ;OK SO, INCREMENT VIDEO
06410         INC     IY            ;AND MEMORY LOCATION
06420         JP      CONT3         ;GO SCAN FOR NEXT INPUT
06430 LFEED1  PUSH    AF            ;SAVE CARRET BYTE
06440         LD      A,1           ;STUFF 1 INTO
06450         LD      (LNFEED),A    ;AUTO LINE FEED POINTER
06460         POP     AF            ;RESTORE CARRIAGE RETURN
06470         RET                   ;RETURN WHENCE U CAME +1
06480 LFEED2  XOR     A             ;ZERO OUT
06490         LD      (LNFEED),A    ;LINEFEED POINTER
06500         LD      A,10          ;ASCII 10 = LINEFEED
06510         JP      CONT5         ;GO STUFF IT IN MEMORY
06520 LNFEED  DEFB    0             ;LINEFEED POINTER STASH
06530 LEFT1   CALL    SLOWLY        ;SLOWDOWN CURSOR MOVEMENT
06540         CALL    CKBACK        ;CHECK IN BOUNDS ?
06550         DEC     IX            ;OK, MOVE BACK A SPACE
06560         DEC     IY            ;AND DOWN 1 MEM LOCATION
06570         JP      CONT3         ;GO SCAN NEXT INPUT
06580 RIGHT1  CALL    SLOWLY        ;SLOWDOWN CURSOR MOVEMENT
06590         CALL    CKAHED        ;CHECK IN BOUNDS ?
06600         INC     IX            ;OK, MOVE AHEAD A SPACE
06610         INC     IY            ;AND UP 1 MEM LOCATION
06620         JP      CONT3         ;GO SCAN NEXT INPUT
06630 UPONE   CALL    SLOWLY        ;SLOWDOWN CURSOR MOVEMENT
06640         CALL    SLOWLY        ;SLOWDOWN CURSOR MOVEMENT
06650         CALL    CKDOWN        ;CHECK IN BOUNDS ?
06660         CALL    SUB64         ;OK, SO MOVE UP A LINE
06670         JP      CONT3         ;GO SCAN NEXT INPUT
06680 DOWN1   CALL    SLOWLY        ;SLOWDOWN CURSOR MOVEMENT
06690         CALL    SLOWLY        ;SLOWDOWN CURSOR MOVEMENT
06700         CALL    CKUP          ;CHECK IN BOUNDS ?
06710         CALL    ADD64         ;OK, SO MOVE DOWN A LINE
06720         JP      CONT3         ;GO SCAN NEXT INPUT
06730 CONT3A  POP     HL            ;ADJUST STACK
06740         JP      CONT3         ;GO SCAN NEXT INPUT
06750 CKBACK  LD      DE,15360      ;BEGIN VIDEO MEMORY
06760         PUSH    IX            ;SWAP IX
06770         POP     HL            ;INTO HL
06780         CALL    0A39H         ;COMPARE HL MINUS DE
06790         JP      Z,CONT3A      ;IF EQUAL, THEN IGNORE
06800         RET                   ;ELSE OK.  RETURN
06810 CKAHED  LD      DE,16383      ;END OF VIDEO MEMORY
06820         PUSH    IX            ;SWAP IX
06830         POP     HL            ;INTO HL
06840         CALL    0A39H         ;COMPARE HL MINUS DE
06850         JP      Z,CONT3A      ;IF EQUAL, THEN IGNORE
06860         RET                   ;ELSE OK.  RETURN
06870 CKDOWN  CALL    SUB64A        ;-64 FROM VIDEO MEM
06880         LD      DE,15360      ;BEGIN VIDEO MEM
06890         CALL    0A39H         ;COMPARE HL - DE
06900         JP      C,CONT3A      ;IF OUT OF BOUNDS, IGNORE
06910         RET                   ;ELSE OK.  RETURN
06920 CKUP    CALL    ADD64A        ;+64 TO VIDEO MEM
```

```
06930         LD     DE,16384      ;END VIDEO MEM
06940         CALL   0A39H         ;COMPARE HL - DE
06950         JP     NC,CONT3A     ;IF OUT OF BOUNDS, IGNORE
06960         RET                  ;ELSE OK.  RETURN
06970 SUB64A  PUSH   IX            ;SWAP IX
06980         POP    HL            ;INTO HL
06990         LD     A,64          ;WE COULD HAVE
07000 AGN64S  DEC    HL            ;USED ADD HL,DE
07010         DEC    A             ;BUT THERE IS MORE
07020         RET    Z             ;THAN ONE WAY TO
07030         JP     AGN64S        ;SKIN A CAT
07040 ADD64A  PUSH   IX            ;SWAP IX
07050         POP    HL            ;INTO HL
07060         LD     A,64          ;WE COULD HAVE
07070 AGN64A  INC    HL            ;USED SBC HL,DE
07080         DEC    A             ;BUT THERE IS MORE
07090         RET    Z             ;THAN ONE WAY TO
07100         JP     AGN64A        ;SKIN A CAT
07110 SUB64   LD     A,64          ;HERE IS ANOTHER
07120 AGNSUB  DEC    IX            ;PLACE YOU MIGHT
07130         DEC    IY            ;WISH TO USE
07140         DEC    A             ;SBC HL,DE
07150         RET    Z             ;HOW MANY BYTES
07160         JP     AGNSUB        ;DID IT SAVE ?
07170 ADD64   LD     A,64          ;HERE IS ANOTHER
07180 AGNADD  INC    IX            ;PLACE YOU MIGHT
07190         INC    IY            ;WISH TO USE
07200         DEC    A             ;ADD HL,DE
07210         RET    Z             ;HOW MANY BYTES
07220         JP     AGNADD        ;DID IT SAVE ?
07230 HOLDIT  DEFW   0             ;HOLDIT STASH
07240 SLOWLY  CALL   BLINKA        ;S   C   M
07250         CALL   BLINKB        ;L   U   0
07260         CALL   BLINKA        ;  0   R   V
07270         CALL   BLINKB        ;   W   S   E
07280         CALL   BLINKA        ;    D   0   M
07290         CALL   BLINKB        ;     0   R   E
07300         CALL   BLINKA        ;      W       N
07310         CALL   BLINKB        ;       N       T
07320         RET                  ;RETURN WHENCE U CAME +1
07330 BLINKA  LD     A,(IX)        ;SAVE VIDEO BYTE
07340         LD     (HOLDIT),A    ;IN HOLDIT
07350         LD     A,143         ;RECTANGULAR CURSOR
07360         LD     (IX),A        ;DISPLAY ON VIDEO
07370         LD     BC,600        ;1/100TH SECOND
07380         CALL   060H          ;TIME DELAY
07390         RET                  ;RETURN WHENCE U CAME +1
07400 BLINKB  LD     A,(HOLDIT)    ;RESTORE VIDEO CHARACTER
07410         LD     (IX),A        ;TO VIDEO MEM LOCATION
07420         LD     BC,600        ;1/100TH SECOND
07430         CALL   060H          ;TIME DELAY
07440         RET                  ;RETURN WHENCE U CAME +1
07450 INVERT  LD     A,(HOLDZ)     ;INVERT UPPER/LOWER CASE
07460         CP     65            ;SUBTRACT 65
07470         JP     M,CONT5       ;NOT ALPHABETICAL IGNORE
07480         CP     123           ;SUBTRACT 123
07490         JP     P,CONT5       ;NOT ALPHABETICAL IGNORE
07500         CP     95            ;SUBTRACT 95
07510         JP     Z,CONT5       ;NOT ALPHABETICAL IGNORE
07520         BIT    5,A           ;BIT 5 SET ?
07530         JP     Z,SET5A       ;IF NOT, THEN SET XT
07540         RES    5,A           ;ELSE RESET IT
07550         JP     CONT5         ;AND DISPLAY IT
07560 SET5A   SET    5,A           ;SET BIT 5 TO DISPLAY
07570         JP     CONT5         ;AND DISPLAY IT
07580 HOLDZ   DEFB   0             ;BYTE STASH
07590 UPSIDE  DEFB   0             ;LOWER CASE POINTER
07600 CK      PUSH   AF            ;SAVE BYTE
07610         LD     A,(14464)     ;SHIFT KEY PRESSED ?
07620         CP     1             ;IF SO
07630         JP     Z,COR         ;IGNORE IT
07640         POP    AF            ;RESTORE BYTE
07650         JP     CONT5         ;CONTINUE ONWARD
07660 COR     POP    AF            ;ADJUST STACK FOR PUSH
07670         JP     CONT3         ;GO SCAN NEXT INPUT
07680 ONE28   CALL   SLOWLY        ;SLOWDOWN AS THIS IS AN
07690         CALL   SLOWLY        ;AUTO REPEAT FUNCTION
07700         LD     A,128         ;END OF MESSAGE DELIMITER
07710         JP     CONT5         ;STUFF IT IN MEM & VIDEO
07720 NOTASC  CP     16            ;ELECTRIC PENCIL CTRL KEY
07730         JP     Z,29760       ;REINITIALIZE PGM POINTER
07740         LD     A,(14352)     ;KYBD ZERO PSUEDO MEMORY
07750         CP     1             ;SHIFT ZERO PRESSED ?
07760         JP     Z,ONE28       ;END OF MESSAGE DELIMITER
07770         LD     A,(14337)     ;KYBD @ PSUEDO MEMORY
07780         CP     1             ;@ KEY PRESSED ?
07790         JP     NZ,CONT3B     ;IF NOT, CONTINUE ONWARD
07800         CALL   CLS           ;C L E A R  V I D E O
07810         CALL   CARETN        ;VIDEO SKIP A LINE
07820         LD     HL,VALMS      ;STACK POINTER MESSAGE
07830         CALL   DIZPLA        ;DISPLAY IT ON VIDEO
07840         LD     HL,0          ;ZERO OUT HL
07850         ADD    HL,SP         ;ADD IT TO STACK VALUE
07860         CALL   0A9AH         ;MOVE IT TO ACCUM
07870         XOR    A             ;ZERO OUT 'A'
07880         CALL   1034H         ;GENERATE
07890         OR     (HL)          ;UNSIGNED
07900         CALL   0FD9H         ;INTEGER
07910         CALL   DIZPLA        ;DISPLAY IT ON VIDEO
07920         CALL   CARETN        ;VIDEO CARRIAGE RETURN
07930         CALL   CARETN        ;VIDEO CARRIAGE RETURN
07940         LD     HL,VALMS0     ;MEMORY LOCATION MESSAGE
07950         CALL   DIZPLA        ;DISPLAY IT ON VIDEO
07960         PUSH   IY            ;SWAP IY MEM LOCATION
07970         POP    HL            ;INTO HL
07980         CALL   0A9AH         ;MOVE HL TO ACCUM
07990         XOR    A             ;ZERO OUT 'A'
08000         CALL   1034H         ;GENERATE
08010         OR     (HL)          ;UNSIGNED
08020         CALL   0FD9H         ;INTEGER
08030         CALL   DIZPLA        ;DISPLAY IT ON VIDEO
08040         CALL   CARETN        ;VIDEO CARRIAGE RETURN
08050         CALL   CARETN        ;VIDEO CARRIAGE RETURN
08060         LD     HL,VALMS1     ;MEM VALUE MESSAGE
08070         CALL   DIZPLA        ;DISPLAY IT ON VIDEO
08080         LD     A,(IY)        ;IY LOCATION MEM VALUE
08090         LD     L,A           ;INTO 'L'
08100         LD     H,0           ;ZERO OUT 'H'
```

```
08110         CALL    0A9AH          ;MOVE HL TO ACCUM
08120         CALL    0FBDH          ;CONVERT ACCUM TO STRING
08130         CALL    DIZPLA         ;AND DISPLAY IT ON VIDEO
08140         CALL    CARETN         ;VIDEO CARRIAGE RETURN
08150         CALL    CARETN         ;VIDEO CARRIAGE RETURN
08160         LD      HL,VALMS2      ;INPUT NEW MEM MESSAGE
08170         CALL    DIZPLA         ;DISPLAY IT ON VIDEO
08180         LD      BC,32000       ;1/2 SECOND
08190         CALL    060H           ;TIME DELAY
08200         CALL    1BB3H          ;INPUT NEW VALU FROM KYBD
08210         RST     10H            ;SCAN STRING SET 'C' FLAG
08220         CALL    0E6CH          ;ASCII $ TO ACCUM RET MIN
08230         CALL    0A7FH          ;CONVERT ACCUM TO INTEGER
08240         LD      A,L            ;NEW MEM VALUE
08250         LD      (IY),A         ;AND STUFF IT IN MEMORY
08260 NOTAS   LD      HL,(LASMEM)    ;BEGINNING MEM LOCATION
08270         LD      DE,15360       ;BEGINNING VIDEO MEM
08280         LD      BC,1024        ;RESTORE VIDEO ALMOST
08290         LDIR                   ;SAME AS BEFORE
08300         CALL    CKAHED         ;TEST VIDEO IN BOUNDS ?
08310         INC     IX             ;OK, SO MOVE CURSOR AHEAD
08320         INC     IY             ;& INCREMENT MEM LOCATION
08330         JP      CONT3          ;GO BACK & SCAN KEYBOARD
08340 VALMS   DEFM    'STACK POINTER = '
08350         DEFB    0              ;DELIMITER
08360 VALMSO  DEFM    'MEM LOCATION IS '
08370         DEFB    0              ;DELIMITER
08380 VALMS1  DEFM    'MEMORY VALUE IS '
08390         DEFB    0              ;DELIMITER
08400 VALMS2  DEFM    'INPUT NEW VALUE '
08410         DEFB    0              ;DELIMITER
08420 MEMO    DEFW    0              ;MEMORY LOCATION STASH
08430 MEMO1   DEFW    0              ;MEM LOCATION STASH -1
08440 NEXT2   LD      BC,24000       ;ABOUT 1/3 SECOND
08450         CALL    060H           ;TIME DELAY
08460         JP      NEXT           ;AWAIT EDIT MODE COMMAND
08470 CARETN  LD      A,13           ;VIDEO
08480         CALL    033H           ;CARRIAGE RETURN
08490         RET                    ;RETURN WHENCE U CAME +1
08500 CLS     LD      HL,15360       ;BEGIN VIDEO MEM
08510         LD      DE,15361       ;PLUS ONE
08520         LD      BC,1023        ;BYTES TO CLEAR
08530         LD      (HL),32        ;WITH SPACES
08540         LD      (16416),HL     ;RESET VIDEO CURSOR
08550         LDIR                   ;MOVE'M RIGHT NOW
08560         RET                    ;RETURN WHENCE U CAME +1
08570 DIZPLA  PUSH    AF             ;SAVE
08580         PUSH    BC             ;EVERTHING
08590         PUSH    DE             ;INCLUDING
08600         PUSH    HL             ;THE
08610         PUSH    IX             ;KITCHEN
08620         PUSH    IY             ;SINK
08630 MORE1   LD      A,(HL)         ;BYTE TO DISPLAY
08640         CP      0              ;END MESSAGE DELIMITER
08650         JP      Z,FINIS1       ;IF SO, ALL DONE
08660         CALL    033H           ;DISPLAY & UPDATE CURSOR
08670         INC     HL             ;MESSAGE MEM LOCATION
08680         JP      MORE1          ;GO DISPLAY NEXT ONE
08690 FINIS1  POP     IY             ;SINK
08700         POP     IX             ;KITCHEN
08710         POP     HL             ;THE
08720         POP     DE             ;INCLUDING
08730         POP     BC             ;EVERTHING
08740         POP     AF             ;RESTORE
08750         RET                    ;RETURN WHENCE U CAME +1
08760
08770 ; - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
08780 ; END OF EDIT/MODIFY/MONITOR SUBROUTINE
```

A MINI-SIZED BULLETIN BOARD SYSTEM

POSSIBLE STANDARD FOR PACKET RADIO

Robert M. Richardson, W4UCH
22 North Lake Drive
Chautauqua Lake, N.Y. 14722

ABSTRACT:

AUTO connect/disconnect mode for unattended operation is available with the Vancouver Area Digital Communications Group% terminal node controller (TNC), the Tucson Amateur Packet Radio TNC, and 'Synchronous Packet Radio Using The Software Approach - AX.25 Protocol' software TNC.

A logical expansion of the auto mode's capabilities would be to allow the station to which your station is connected in the AUTO mode to have full access to one or more of your disk drives. Minimum functions would include: LIST the disk directory, SEND a given disk file, SAVE a given file on disk, and send a set of operating instructions upon receiving the HELP command.

It is obvious that a disk I/O system subroutine such as this in the AUTO mode, is indeed in essence a mini-sized version of a computer bulletin board system. By the term 'Possible Standard* in this paper's title, we are suggesting that it would be a 'nice to have feature' incorporated in all packet stations regardless of the protocol or TNC used.

This paper describes the subroutines used by the author to provide these functions on a Model I TRS-80 with the packet radio software approach using the Vancouver protocol.

GENERAL:

These subroutines were written and tested during the summer of 1983. Why did they use the Vancouver protocol? Quite simply because at that time in the western New York and southern Ontario regions there were no AX.25 stations (other than the author's software approach) on the 2 meter band. Southern Ontario (in the vicinities of Hamilton and Toronto) had about 50 active packeteers, all using the Vancouver TNC with Vancouver protocol, and the Buffalo, NY area 65 miles northeast of our QTH had Gil Boelke - W2EUP, using the Vancouver protocol with the GLB PK-1 software approach.

W2EUP convinced us of the value of implementing disk I/O in the AUTO mode with a number of demonstrations, so we wrote the following subroutines. They are designed to work only with the following Model I TRS-80 disk operating systems: TRSDOS 2.3, NEWDOS + and NEWDOS 1.0.

Figure 1 is the commented source code for this subroutine. The comments are largely self explanatory. The equates (EQU) at the beginning of the program serve to link these AUTO mode subroutines to the main software program used in Volume 3 of ‘Synchronous Packet Radio Using The Software Approach - Advanced Vancouver Protocol.’

Though only the SEND, SAVE, LIST, and HELP commands are used in this mini-subroutine it is a relatively simple matter to expand the commands to include FLAGS xxx to set the program's number of opening flags transmitted, DELETE file name to do just that, and to include the disk drive number with the file name so any number of disk drives from 1 to 4 may be accessed.

Depending upon how far you wish to go, this fundamental program may be expanded up to and including all of the features of a full sized computer bulletin board system.

Originally, the AUTO disk I/O subroutines required a carriage return and line feed immediately after each command to eliminate the possibility of having the program confuse an info field in a frame that began with SEND, SAVE, LIST or HELP as a command rather than part of the message. Since the commands must be in capital letters, this has not occurred during the past 9 months of operation so the mandatory carriage return and line feed requirements were removed to further simplify operation. By all means put them back in if you wish.

W2EUP's and K2IMF's AUTO mode disk I/O programs require that each command begin with the / character; i.e., /SEND, /SAVE, /LIST, and /HELP to avoid possible confusion. This is yet another approach you might consider.

Figure 2 is the HELP message the program sends in the AUTO mode upon receiving the HELP command in a single frame packet after the connection is established. The capital M's represent ASCII 13 carriage returns and the capital

3% represent ASCII 10 line feeds. They are included, as some of the dumb terminals now used with some packet TNCs do not have the automatic scrolling feature.

CONCLUSION:

Expanding the auto connnect mode's capability to include disk I/O has proven extremely useful. We are grateful to W2EUP for the concept and for continued encouragement.

Volume 3 of @Synchronous Packet Radio Using The Software Approach - Advanced Vancouver Protocol' has not been published and probably never will be published.

WHY?

Quite simply the brilliant AX.25 protocol is taking the world by storm. Its growth capabilites and myriad other advantages over the older Vancouver protocol have just about buried the old timer once and for all except in the immediate Toronto, Vancouver, and Melbourne environs.

The other reason 'Advanced Vancouver Protocol' may never be published is simply the costs involved. It takes a minimum of 200 sales per volume just to break even and with the greatest stretch of our wildest imagination we cannot forsee 200 knowledgeable radio amateurs springing for yet another Vancouver protocol book.

If you feel you must have a copy of the uncommented source and object code for Volume 3, available only for the Model I TRS-80, on a single, DOUBLE sided, 35 track disk, with no instructions or assistance whatsoever other than a short single page telling you how to install your own call letters and node number in the program, then send Richcraft a check for $29 and the disk will be mailed to you first class.

Richcraft Engineering Ltd.
#1 Wahmeda Industrial Park
Chautauqua, New York 14722

NOTE:
The main and shift menus of the Volume 3 program are similar to the menus illustrated in Figure 2 of the Ax.25 Protocol paper by the same author that is presented elsewhere in these proceedings.

```
VOLUME 3 MAIN MENU;

                    ENTER OPTION DESIRED 3 _

CHANGE ADDRESSEE CALL        = A   W2EUP CONNECT REQUEST CQ = B
NOT CONNECTED TOGGLE         = C   W2EUP DISCONNECT REQUEST = D
SEND PACKETS FROM LO-MEM     = E   W2EUP CONNECT ACKNOWLEDGE = F
INPUT FRAMES/PACKET LO-MEM   = G   THIS IS VANCOUVER PROTOCOL = H
BACKOFF DELAY TOGGLE OFF     = I   AUTO CONNECT TOGGLE OFF   = J
NOW IN UPPER CASE MODIFY     = K   W2EUP - GIL BOELKE MESSAGE = L
DISPLAY/EDIT MEMORY PAGE     = M   SET INFO FIELD LOMEM PACKS = N
NOT FORMAT VIDEO TOGGLE      = O   QUICK BROWN FOX MESSAGE   = P
TRANSMIT EXTERNALLY ONLY     = Q   SET OPENING FLAG LENGTH   = R
TRANSMIT TO HI-MEM ONLY      = S   INPUT/XMIT NORMAL INFO = V & T
CLEAR NON-PGM MEM 17K-62K    = U   INPUT/XMIT ALL STATION = V 6 W
ABORT LOW-MEY PAK SEQUENCE   = X   SET RF-TRY IN CONNECT MODE = Y
SHIFT MENU                   = 1   MOVE HI-MEM TO LOW-MEM    = 2
SEND WAIT REQUEST (RNR)      = 3   SEND CLEAR WAIT (RR)      = 4
```

```
VOLUME 3 SHIFT MENU:

                    SHIFT MENU ? _

XMIT 40960 UP CONTINUOUSLY = A   BOOT DOS READY               = B
LOAD HI-MEM ASCII UUUUUUU  = C   LOAD HI-MEM LOGIC 1111111    = D
EDIT/MODIFY INSTRUCTIONS   = E   CHANGE RECEIVE DPLL BASE #   = F
LOG ON VE3MHZ REPEATER     = G   LOG OFF VE3MHZ REPEATER      = H
SEND MORSE I.D.            = I   SEND SEQUENTIAL ACKS         = J
CAUTION ** RESTORE DOS **  = K   DISPLAY LOW MEMORY @ 17408   = L
DISPLAY RCCV PACKS @ 53248 = M   RESTORE PROGRAM POINTERS     = N
DISPLAY CALL/ADDRESS LIST  = O   MOVE PROGRAM TO LOW MEMORY   = P
SAVE HI-MEM ON DISK        = Q   LOAD DISK FILE TO HI-MEM     = R
TRANSMIT BAUD RATE SELECT  = S   SEND DISK :1 DIRECTORY       = T
CLEAR HI-MEMORY 53248 +    = U   RECEIVE VANCOUVER PROTOCOL   = V
RECEIVE AX.25 NOT CONNECT  = W   SEND MORSE FROM KEYBOARD     = X
NORMAL DISPLAY - NOT DPLL  = Y   DISPLAY DPLL LAST QUADRANT   = Z
      NOTE: SPACE BAR IN RECEIVE MODE = RESEND LAST PACK
```

```
VOLUME 3 CALL & ADDRESS LIST (SUMMER '83):

                 - CALL AND ADDRESS LIST -

VE7APU = 196    VE3ATI =  51    VE2BAR = 215    VE3BKB = 220
 (DOUG)          (BERNIE)       (MIKE)          (JON)
VE3DNM = 98     VE3DSP =  97    VE3DVV = 115    VE3DRZ = 239
(MAX)           (GLEN)          (JOHN)          (ROB)
VE3EC = 236     VE3EHL = 117    W2EUP  = 119    VE3FAO = 250
 (BILL)          (ED)            (GIL)          (FRANK)
VE3FGK = 218    VE3FMG = 120    VE3GBC = 216    VE3IAC = 238
 (DAVE)          (MIKE)          (BRUCE)         (PAUL)
VE3IUV = 116    VE3LNY = 181    VE3MWM = 185    VE3NEC = 210
(RON)            (JACK)         (STU)           (JOHN)
VE3PKT = 186    WA2RYT = 129    VE3SP  = 118    VE3UR = 221
 (MAIL BOX)     (RAY)            (RON)          (RAY)
K2IMF  = 41     WB2VEU =  65    W4UMF  =  28    W2CIX  = 30
 (DON)           (ANDY)         (TOM)            (BILL)
```

```
POTENTIAL OSCAR 10 PACKETEERS (SUMMER '83):

U.S.A.   :  - POTENTIAL OSCAR 10 PACKET LIST AS OF 8/15/83 -
KA1GD  ANDY     KA1HTV UNK      WA1LOU STAN     W2EUP  GIL
K2IMF  DON      WA2 LQQ GRUM    WB2VEU ANDY     W3IWI  TOM
K4BRK  CARL     K4CAV  CHAS     WB4JFI TERRY    W4RI   PAUL
W4UCH  BOB      W4UMF  TOM      WA6JPR WALT     NK6K   HAL
KA6M   HANK     W6OVP  DAVE     W6TNS  DON      WA7GXD LYLE
W8KOX  TOM      W9BD   FRED     WB9FLW PETE     K9NG   STEVE
KA9NZI GARY     KA9Q   PHIL     NOCCZ  ANDY     KROU   TIM
OVERSEAS:
OK2SPS PETER    SM5HEV JENS     VK2BOA TONY     ZL1AOX IAN
CANADA  :
VE2BAR MIKE     VE2BPD JEAN     VE3ATI BERNIE   VE3BKB JON
VE3DNM MAX      VE 3DRZ ROB     VE3DSP GLEN     VE3DVV JOHN
VE3EC  BILL     VE3EHL ED       VE3FAO FRANK    VE3FGK DAVE
VE3FMG MIKE     VE3GBC BRUCE    VE3IAC PAUL     VE3IUV RON
VE3LNY JACK     VE3MWM STU      VE3NEC JOHN     VE3SP  RON
```

FIGURE 1

```
00100 ; AUTO DISK I/O SUBROUTINES FOR VOLUME 3
00110
00120 ; SYNCHRONOUS PACKET RADIO USING THE SOFTWARE APPROACH
00130
00140 ; FOR TRSDOS 2.3 - NEWDOS PLUS - NEWDOS80 1.0
00150
00160 ; SOURCE - AUTO 1                    OBJECT - AUTO 2
00170
00180 ; COPYRIGHT (C) 1984 BY RICHCRAFT ENGINEERING LTD.
00190
00200 SETUP   EQU     29760           ;INITIALIZE PGM POINTERS
00210 LENGTH  EQU     75FBH           ;INFO FIELD LENGTH/FRAME
00220 AUTSAV  EQU     75FDH           ;AUTO SAVE DISK POINTER
00230 CONREQ  EQU     75FEH           ;SABM POINTER
00240 BEFORE  EQU     7601H           ;END RCVD INFO III-MEMORY
00250 SEGNUM  EQU     7603H           ;VERY LONG FILE SEGMENTS
00260 CLRHY   EQU     7604H           ;CLEAR HI-MEMORY
00270 SEND3Z  EQU     7607H           ;XMIT MESSAGE ADDRESS
00280 PACK    EQU     760AH           ;XMIT MULTI-FRAME ADDRESS
00290 FRAMES  EQU     7611H           ;NUMBER OF FRAMES/PACK
00300 MOVDN   EQU     761FH           ;RESTORE DOS FM MID-MEM
00310 SIGN9   EQU     762FH           ;TYPE FUNCTION AUTO PTR
00320 MENU    EQU     7630H           ;DISPLAY MAIN MENU
00330         ORG     49632           ;DISK FILE CONTROL BLOCK
00340 FCB     DEFS    32              ;SAVE 32 BYTES FOR FCB
00350 BUFFER  DEFS    256             ;DISK I/O WORKING SPACE
00360 CHEKIT  INC     HL              ;TEST
00370         LD      A,(HL)          ;THE
00380         CP      'L'             ;AUTO
00390         JP      Z,TESTL         ;FUNCTION
00400         CP      'H'             ;DESIRED
00410         JP      Z,TESTH         ;BY
00420         CP      'S'             ;THE
00430         JP      Z,TESTS         ;STATION
00440         RET                     ;WHO
00450 TESTL   INC     HL              ;CONNECTED
00460         LD      A,(HL)          ;TO
00470         CP      'I'             ;YOUR
00480         RET     NZ              ;STATION
00490         INC     HL              ;WHICH
00500         LD      A,(HL)          ;IS
00510         CP      'S'             ; NOW
00520         RET     NZ              ;IN
00530         INC     HL              ;THE
00540         LD      A,(HL)          ;AUTO
00550         CP      'T'             ;MODE.
00560         RET     NZ              ;GOT A 'LIST' COMMAND ?
00570         LD      A,8             ;IF SO,
00580         LD      (SIGN9),A       ;THEN SET SIGN9 FOR LIST
00590         RET                     ;RETURN WHENCE U CAME +I
00600 TESTH   INC     HL              ;LAST LETTER WAS AN 'H'
00610         LD      A,(HL)          ;SO
00620         CP      'E'             ;TEST
00630         RET     NZ              ;FOR
00640         INC     HL              ;THE
00650         LD      A,(HL)          ;'H'
00660         CP      'L'             ;'E'
00670         RET     NZ              ;'L'
00680         INC     HL              ;'P'
00690         LD      A,(HL)          ;COMMAND
00700         CP      'P'             ;WAS IT 'HELP' ?
00710         RET     NZ              ;IF NOT, IGNORE IT
00720         LD      A,2             ;IF SO,
00730         LD      (SIGN9),A       ;THEN SET SIGN9 FOR HELP
00740         RET                     ;RETURN WHENCE U CAME +1
00750 TESTS   INC     HL              ;LAST LETTER WAS AN 'S'
00760         LD      A,(HL)          ;TEST FOR
00770         CP      'A'             ;'A' AS IN SAVE
00780         JP      Z,TESTA         ;IF SO, TEST FOR 'V'
00790         CP      'E'             ;ELSE TEST FOR 'E' AS IN
00800         JP      Z,TESTE         ;SEND. IF SO, TEST 'N'
00810         RET                     ;RETURN WHENCE U CAME +1
00820 TESTA   INC     HL              ;GOT AN 'S' AND 'A'
00830         LD      A,(HL)          ; NOW
00840         CP      'V'             ;TEST
00850         RET     NZ              ;FOR
00860         INC     HL              ;A 'V'
00870         LD      A,(HL)          ;AND
00880         CP      'E'             ;'E'
00890         RET     NZ              ;IF SO,
00900         INC     HL              ;ALSO TEST
00910         LD      A,(HL)          ;FOR A
00920         CP      ' '             ;SPACE AFTER SAVE
00930         RET     NZ              ;IF NOT,
00940         INC     HL              ;THEN IGNORE IT
00950         LD      A,6             ;ELSE
00960         LD      (SIGN9),A       ;SET SIGN9 FOR 'SAVE'
00970         LD      (SAVEIT),HL     ;AND SAVE MEM LOCATION
00980         RET                     ;FOR 'SAVE' FILE NAME
00990 TESTE   INC     HL              ;GOT AN 'S' AND 'E'
01000         LD      A,(HL)          ;SO
01010         CP      'N'             ;TEST
01020         RET     NZ              ;FOR
01030         INC     HL              ;'N'
01040         LD      A,(HL)          ;AND
01050         CP      'D'             ;A
01060         RET     NZ              ;'D' ?
01070         INC     HL              ;AND
01080         LD      A,(HL)          ;A
01090         CP      ' '             ;SPACE ?
01100         RET     NZ              ;IF
01110         INC     HL              ;SO ?
01120         LD      A,4             ;THEN SET
01130         LD      (SIGN9),A       ;SIGN9 FOR A 'SEND'
01140         LD      (SAVEIT),HL     ;AND SAVE MEM LOCATION
01150         RET                     ;FOR SEND FILE NAME
01160 DIZ     LD      A,(HL)          ;DISPLAY
01170         CP      0               ;A MESSAGE
01180         JP      Z,FINISH        ;ON VIDEO
01190         CALL    033H            ;AT CURSOR
01200         INC     HL              ;LOCATION WITH
01210         JP      DIZ             ;ZERO DELIMITER
01220 DZ      CALL    0A9AH           ;MOVE HL TO ACCUM
01230         CALL    0FBDH           ;CONVERT ACCUM TO STRING
01240         CALL    DIZ             ;DISPLAY IT ON VIDEO
01250         RET                     ;RETURN WHENCE U CAME +1
01260 FINISH  RET                     ;RETURN WHENCE U CAME +1
01270 CLS     LD      HL,15360        ;C L E A R   V I D E O
```

```
01280          LD     (16416),HL       ;RESET VIDEO CURSOR
01290          LD     DE,15361         ;CLEAR THE
01300          LD     BC,1023          ;1024 BYTE
01310          LD     (HL),32          ;VIDEO DISPLAY
01320          LDIR                    ;WITH
01330          RET                     ;SPACES
01340  ERR     DEFM   'SP ERROR '      ;STACK POINTER
01350          DEFB   0                ;ERROR MESSAGE
01360  TESTSP  LD     HL,0             ;THE OPTIONAL
01370          ADD    HL,SP            ;STACK
01380          LD     DE,29756         ;POINTER
01390          RST    18H              ;TEST
01400          RET    Z                ; IS INCLUDED
01410          PUSH   HL               ;FOR TROUBLE
01420          CALL   CLS              ;SHOOTING
01430          LD     HL,ERR           ;YOUR
01440          CALL   DIZ              ;PROGRAM
01450          POP    HL               ;MODIFICATIONS
01460          CALL   DZ               ;SHOULD
01470          CALL   049H             ;YOU
01480          LD     SP,29758         ;WISH
01490          JP     MENU             ;TO USE IT.
01500  INPNAM  CALL   CLS              ;NOT USED BY AUTO MODE
01510          LD     HL,NAM1          ;ONLY FOR KEYBOARD INPUT
01520          CALL   DIZ              ;REMEMBER DELIMITERS MSG?
01530          CALL   049H             ;AWAIT KEYBOARD INPUT
01540          CP     1                ;BREAK KEY PRESSED ?
01550          JP     Z,ESCAPE         ;IF SO, GOT0 MAIN MENU
01560          CALL   CLS              ;C L E A R   V I D E O
01570          LD     HL,NAM1A         ;BEGIN ADDRESS MESSAGE ?
01580          CALL   DIZ              ;ON VIDEO
01590          CALL   1BB3H            ;INPUT BEGIN ADDRESS
01600          RST    10H              ;SCAN STRING SET 'C' FLAG
01610          CALL   1E5AH            ;CONVERT UNSIGNED INTEGER
01620          EX     DE,HL            ;SWAP INTO HL REGISTER
01630          LD     (DUMP+1),HL      ;LOAD BEGIN INTO DUMP
01640          LD     (HOWFAR+1),HL    ;AND BEGIN INTO HOWFAR
01650  INNAME  CALL   CLS              ;C L EA R   V I D E O
01660          LD     HL,NAM2          ;INPUT FILE NAME MESSAGE?
01670          CALL   DIZ              ;ON VIDEO
01680          CALL   1BB3H            ;INPUT FILE NAME
01690          LD     HL,41E8H         ;WHERE STASHED IN MEMORY
01700          LD     A,(HL)           ;TEST FOR NOTHING INPUT
01710          CP     0                ;IF SO,
01720          JP     Z,ESCAPE         ;GOTO MAIN MENU
01730          CALL   LONG             ;HOW MANY BYTES IN NAME?
01740          LD     HL,41E8H         ;NAME LOCATION IN MEMORY
01750          LD     DE,FCB           ;SO MOVE THEM TO
01760          LDIR                    ;FILE CONTROL BLOCK
01770          CALL   DRIVE            ;AND DRIVE NUMBER TOO
01780          RET                     ;RETURN WHENCE U CAME +1
01790  ESCAPE  POP    AF               ;ADJUST STACK FOR CALL
01800          LD     HL,53248         ;RESET TO NORMAL
01810          LD     (DUMP+1),HL      ;DUMP AND
01820          LD     (HOWFAR+1),HL    ;HOWFAR
01830          JP     MENU             ;GOTO MAIN TIP MENU
01840  LONG    LD     BC,0             ;HOW LONG IS FILE NAME ?
01850  LON1    LD     A,(HL)           ;BYTE FROM NAME STRING
01860          CP     0                ;ZERO DELIMITER
01870          RET    Z                ;RETURN WITH BYTE COUNT
01880          INC    C                ;1 MORE BYTE
01890          INC    HL               ;NEXT MEM LOCATION
01900          JP     LON1             ;GO COUNT IT
01910          ORG    0C440H           ;LIST SUBROUTINE LOCATION
01920  LIST1   XOR    A                ;ZERO OUT LIST POINTER
01930          LD     (SIGN9),A        ;AT BEGIN RECEIVE MODE
01940          LD     A,195            ;RESTORE JUMP
01950          LD     (400CH),A        ;IN LOW MEMORY
01960          CALL   MOVDN            ;RESTORE DOS
01970          LD     HL,GGON          ;COMPOSITE DIRECTORY SUB-
01980          LD     (4468H),HL       ;ROUTINE SEARCH ADDRESS
01990          CALL   CLS              ;FOR TRSDOS 2.3 AND
02000          LD     HL,MS1A          ;NEWDOS + AND
02010          CALL   DIZ              ;NEWDOS 1.0
02020          LD     HL,MS1A          ;COMPATIBILITY
02030          JP     4405H            ;DISPLAY DIRECTORY VIDEO
02040  GOON    PUSH   HL               ;SINCE
02050  GOON1   LD     A,(HL)           ;TRSDOS 2.3
02060          CP     3                ;AND
02070          JR     Z,DUN            ;NEWDOS +
02080          CP     'D'              ;AND
02090          JP     Z,TESDOS         ;NEWDOS 1.0
02100  GOON2   CALL   033H             ;USE
02110          INC    HL               ;SOMEWHAT
02120          CP     13               ;DIFFERENT
02130          JR     NZ,GOON1         ;LIST THE
02140  DUN     POP    HL               ;DIRECTORY
02150          RET                     ;SUBROUTINES,
02160  TESDOS  PUSH   HL               ;THIS
02170          LD     (HOLE),A         ;SIMPLE
02180          INC    HL               ;PROGRAM
02190          LD     A,(HL)           ;FINESSES
02200          CP     '0'              ;THE
02210          JP     NZ,DUN1          ;PROBLEM
02220          INC    HL               ;BY
02230          LD     A,(HL)           ;FIRST
02240          CP     'S'              ;DISPLAYING
02250          JP     NZ,DUN1          ;THE
02260          POP    HL               ;DIRECTORY
02270          POP    DE               ;ON
02280          POP    BC               ;VIDEO.
02290          JP     DUN2             ;THEN,
02300  HOLE    DEFB   0                ;THE
02310  DUN1    LD     A,(HOLE)         ;LENGTH
02320          POP    HL               ;OF
02330          JP     GOON2            ;THE
02340  DUN2    LD     HL,(16416)       ;DIRECTORY
02350          DEC    HL               ; ON
02360          DEC    HL               ;THE
02370          DEC    HL               ;VIDEO
02380  GOBAK   LD     A,(HL)           ;DISPLAY
02390          CP     47               ;IS
02400          JP     P,ONWARD         ;CALCULATED
02410          DEC    HL               ;USING
02420          JP     GOBAK            ;THE
02430  ONWARD  INC    HL               ;* DOS
02440          INC    HL               ;READY
02450          EX     DE,HL            ;MESSAGE,
```

```
02460          LD      HL,15360         ;AND
02470          CALL    0BC7H            ;THE
02480          LD      (LBYTES),HL      ;DISK
02490          PUSH    HL               ;DIRECTORY
02500          LD      BC,32000         ;MOVED
02510          CALL    060H             ;FROM VIDEO
02520          POP     BC               ;TO
02530          LD      HL,15360         ;HI-MEMORY
02540          LD      DE,53248         ;RIGHT
02550          LDIR                     ;HERE.
02560          LD      SP,29758         ;RESET STACK POINTER
02570          CALL    SETUP            ;INITIALIZE PGM POINTERS
02580          CALL    CLS              ;CL E AR   V I D E O
02590          CALL    CLRLO            ;CLEAR OUT DOS LO-MEM
02600          CALL    MOVHI1           ;MOVE FROM HI TO LO-MEM
02610          CALL    CALFRM           ;CALCULATE FRAMES/PACK
02620          JP      PACK             ;AND GO SEND THEM.
02630 LBYTES   DEFW    0                ;NUMBER BYTES READ STASH
02640 CLRLO    LD      HL,16872         ;CLEAR OUT DOS
02650          LD      DE,16873         ;FROM
02660          LD      BC,12878         ;LOW MEMORY
02670          LD      (HL),0           ;AND REPLACE
02690          LDIR                     ;WITH
02690          RET                      ;ZEROS.
02700 NUMB     DEFB    0                ;BYTES PER LINE STASH
02710 MOVHI1   LD      A,62             ;USE 62 PER LINE
02720          LD      (NUMB),A         ;AND SET NUMB
02730          LD      HL,53248         ;BEGIN HI-MEMORY
02740          LD      DE,17408         ;BEGIN MULTI-FRAME XMIT
02750          LD      BC,(LBYTES)      ;NUMBER BYTES TO MOVE
02760 MOV1     LD      A,(HL)           ;BYTE FROM HI-MEM
02770          LD      (DE),A           ;MOVE TO LO-MEM
02780          INC     HL               ;NEXT HI-MEM LOCATION
02790          INC     DE               ;NEXT LO-MEM LOCATION
02800          LD      A,(NUMB)         ;BYTES PER LINE
02810          DEC     A                ;MINUS ONE
02820          LD      (NUMB),A         ;AND SAVE IT
02830          CALL    Z,LFEED          ;ZERO DO CARRET/LINEFEED
02840          DEC     BC               ;BYTES TO MOVE COUNTER
02850          LD      A,B              ;TEST
02860          OR      C                ;FOR ZERO ?
02870          JP      Z,STORDE         ;SAVE END LOCATION LO-MEM
02880          JP      MOV1             ;ELSE MOVE NEXT ONE
02890 STORDE   CALL    READY            ;<READY> MSG ON THE END
02900          LD      (THEND),DE       ;SAVE END LO-MEM LOCATION
02910          RET                      ;RETURN WHENCE U CAME +1
02920 THEND    DCFW    0                ;END LOCATION STASH
02930 LFEED    LD      A,62             ;BYTES PER LINE
02940          LD      (NUMB),A         ;RESET COUNTER
02950          LD      A,13             ;CARRIAGE RETURN
02960          LD      (DE),A           ;STUFF INTO LO-MEM
02970          INC     DE               ;NEXT LO-MEM LOCATION
02980          LD      A,10             ;LINE FEED
02990          LD      (DE),A           ;STUFF INTO LO-MEM
03000          INC     DE               ;NEXT LO-MEM LOCATION
03010          RET                      ;RETURN WHENCE U CAME +1
03020 NUM2     DEFB    0                ;NUMBER FRAMES COUNTER
03030 CALFRM   LD      DE,(THEND)       ;LO-MEM END MSG LOCATION
03040          LD      HL,17408         ;BEGIN MULTI-FRAME ADRESS
03050          CALL    0BC7H            ;SUBTRACT HL FROM DE
03060          LD      DE,250           ;250 BYTES ?
03070          CALL    1C90H            ;COMPARE HL - DE
03080          JP      C,SET1           ;< 250 SO SET 1 FRAME
03090          XOR     A                ;ELSE
03100          LD      (NUM2),A         ;RESET COUNTER
03110 NUM1     LD      DE,(LENGTH)      ;INFO FIELD LENGTH/PACK
03120          OR      A                ;CLEAR CARRY FLAG
03130          SBC     HL,DE            ;SUBTRACT DE FROM HL
03140          LD      A,(NUM2)         ;FRAME COUNTER
03150          INC     A                ;PLUS ONE
03160          LD      (NUM2),A         ;AND SAVE IT
03170          JP      Z,FIN            ;IF ZERO ALL DONE
03180          JP      C,FIN            ;IF CARRY ALL DONE
03190          JP      NUM1             ;ELSE DO NEXT ONE
03200 FIN      LD      A,(NUM2)         ;FRAMES COUNT
03210          CP      7                ;7 FRAMES ?
03220          JP      P,SET7           ;MORE THAN 7, SET AT 7
03230          LD      (FRAMES),A       ;ELSE SET AT NUMBER
03240          RET                      ;RETURN WHENCE U CAME +1
03250 SET7     LD      A,7              ;FRAMES/PACK = 7
03260          LD      (FRAMES),A       ;SET FRAME COUNTER
03270          RET                      ;RETURN WHENCE U CAME +1
03280 SET1     LD      A,1              ;FRAMES/PACK = 1
03290          LD      (FRAMES),A       ;SET FRAME COUNTER
03300          RET                      ;RETURN WHENCE U CAME +1
03310          ORG     0C580H           ;HELP MESSAGE SUBROUTINE
03320 HELP1    XOR     A                ;ZERO OUT HELP POINTER
03330          LD      (SIGN9),A        ;AT BEGIN RECEIVE MODE
03340          LD      A,195            ;RESTORE JUMP
03350          LD      (400CH),A        ;IN LOW MEMORY
03360          CALL    MOVDN            ;RESTORE DOS LO-MEM
03370          LD      HL,HELP1A        ;HELP:1 DISK FILE NAME
03380          LD      DE,FCB           ;FILE CONTROL BLOCK
03390          LD      BC,7             ;FILE NAME + DELIMITER
03400          LDIR                     ;MOVE TO FCB
03410          CALL    OPEN1            ;OPEN AN EXISTING FILE
03420          CALL    MULPLY           ;CALCULATE BYTES IN FILE
03430          CALL    READ             ;READ THEM FROM DISK
03440          CALL    CLOSE            ;CLOSE THE DISK FILE
03450          LD      SP,29758         ;RESET STACK POINTER
03460          CALL    SETUP            ;INITIALIZE PGM POINTERS
03470          CALL    CLRLO            ;CLEAROUT DOS FROM LO-MEM
03480          CALL    MOVHI2           ;MOVE FILE HI TO LO-MEM
03490          CALL    CALFRM           ;CALCULATE FRAMES/PACK
03500          CALL    CLRHY            ;CLEAR OUT HI-MEM
03510          JP      PACK             ;SEND MULTI-FRAME PACKS
03520 HELP1A   DEFM    'HELP: 1'        ;HELP FILE NAME AND DRIVE
03530          DEFB    13               ;FCB DELIMITER
03540 OPEN1    LD      DE,FCB           ;OPEN AN EXISTING FILE
03550          LD      HL,BUFFER        ;DISK I/O BUFFER ADDRESS
03560          LD      B,0              ;BYTES PER SECTOR
03570          CALL    4424H            ;DOS OPEN IT SUBROUTINE
03580          JR      NZ,ERROR         ;Z FLAG SET IF ERROR
03590          RET                      ;RETURN WHENCE U CAME +1
03600 MULPLY   LD      A,(FCB+12)       ;NUMBER RECCRDS IN FILE
03610          CP      47               ;MORE THAN 47 ?
03620          JP      P, VYLONG        ;IF SO, GOT0 VERY LONG
03630 MUL0     LD      HL,0             ;BYTES IN FILE COUNTER
```

```
MUL1    LD      DE,256          ;BYTES PER SECTOR IN FILE
        ADD     HL,DE           ;ADD TO BYTE COUNTER
        DEC     A               ;MINUS ONE RECORD
        JP      Z,MUL2          ;IF ZERO, DO LAST ONE
        JP      MUL1            ;ADD UP NEXT ONE
MUL2    LD      A,(FCB+8)       ;BYTES IN LAST SECTOR
        LD      E,A             ;SWAP IN TO 'E'
        LD      D,0             ;ZERO OUT 'D'
        ADD     HL,DE           ;ADD TO TOTAL BYTES
MuL3    LD      (LBYTES) ,HL    ;AND SAVE THEM HERE
        LD      DE,53248        ;BEGIN HI-MEM LOCATION
        ADD     HL,DE           ;HL+DE IN END LOCATION
        LD      (LONG1+1),HL    ;AND SAVE THEM HERE
        RET                     ;RETURN WHENCE U CAME +1
READ    LD      HL,53248        ;WHERE TO PUT FILE IN MEM
        LD      DE,FCB          ;FILE CTRL BLOCK ADDRESS
LG      PUSH    HL              ;SAVE MEM LOCATION STACK
        CALL    13H             ;READ BYTE FROM DISK FILE
        POP     HL              ;RESTORE MEM LOCATION
        LD      (HL),A          ;DISK BYTE INTO MEMORY
        INC     HL              ;NEXT MEM LOCATION
        PUSH    HL              ;SAVE IT ON STACK
        PUSH    DE              ;SAVE FCB POINTER
LONG1   LD      DE,65535        ;MEM END ADDRESS OF FILE
        OR      A               ;CLEAR CARRY FLAG
        SBC     HL,DE           ;SUBTRACT DE FROM HL
        POP     DE              ;RESTORE FCB POINTER
        POP     HL              ;RESTORE MEM LOCATION
        RET     2               ;RETURN IF ALL DONE
        JP      LG              ;READ NEXT DISK FILE BYTE
CLOSE   LD      DE,FCB          ;FCB MEM LOCATION
        CALL    4428H           ;CLOSE FILE SUBROUTINE
        PUSH    AF              ;SAVE FLAG ON STACK
        LD      HL,53248        ;BEGIN HI-MEM ADDRESS
        LD      (DUMP+1),HL     ;RESET DUMP
        LD      (HOWFAR+1),HL   ;RESET HOWFAR
        POP     AF              ;RESTORE FLAG
        RET     2               ;RETURN UNLESS ERROR
        POP     HL              ;ADJUST STACK FOR CALL
ERROR   LD      H,0             ;ZERO OUT 'H'
        LD      L,A             ;ERROR NUMBER TO 'L'
        CALL    0A9AH           ;MOVE HL INTO ACCUM
        CALL    0A7FH           ;MAKE SURE AN INTEGER
        CALL    OFBDH           ;CONVERT ACCUM TO STRING
        LD      DE,MS2A         ;ERROR MESSAGE IN MEM
ER1     LD      A,(HL)          ;FIRST ERROR NUMBER
        CP      0               ;ZERO DELIMITER
        JP      Z,ER2           ;ALL DONE ? GOT0 ER2
        LD      (DE),A          ;ERROR NUMBER TO MESSAGE
        INC     HL              ;NEXT ERROR # LOCATION
        INC     DE              ;NEXT MESSAGE LOCATION
        JP      ER1             ;GO MOVE NEXT ONE
ER2     CALL    CLS             ;C L E A R   V I D E O
        POP     AF              ;ADJUST STACK FOR CALL
        CALL    SETUP           ;INITIALIZE PGM POINTERS
        CALL    CLRLO           ;CLEAR OUT DOS LO-MEM
        CALL    CLRHY           ;CLEAR OUT HI-MEM
        LD      IY,MS2C         ;ERROR # MESSAGE
        JP      SEND32          ;TRANSMIT ERROR # MESSAGE
```

```
MOVH12  LD      HL,53248        ;MOVE HI-MEM TO LO-MEM
        LD      DE,17408        ;WITHOUT CARRET/LINEFEEDS
        LD      BC,(LBYTES)     ;NUMBER OF BYTES TO MOVE
        LDIR                    ;DO IT
        LD      A,128           ;MULTI-FRAME
        LD      (DE),A          ;TRANSMIT
        INC     DE              ;SUBROUTINE
        LD      (DE),A          ;USES
        INC     DE              ;THREE
        LD      (DE),A          ;EACH 128
        INC     DE              ;DELIMTERS
        LD      (THEND),DE      ;IN A ROW
        RET                     ;RETURN WHENCE U CAME +1
        ORG     0C680H          ;AUTO SEND FILE ROUTINE
SEND1   XOR     A               ;ZERO OUT SEND POINTER
        LD      (SIGN9),A       ;AT BEGIN RECEIVE MODE
        LD      HL,(SAVEIT)     ;FILE NAME MEM LOCATION
        LD      DE,FCB          ;FILE CTRL BLOCK ADDRESS
        LD      BC,0            ;ZERO OUT 'BC'
SEN1    LD      A,(HL)          ;MOVE FILE NAME TO FCB
        CP      13              ;CARRIAGE RETURN DELIMTER
        JP      2,SEN2          ;GOTO SEN2 IF DONE
        LD      (DE),A          ;LOAD NAME INTO FCB
        INC     HL              ;NEXT MEM LOCATION
        INC     DE              ;NEXT FCB LOCATION
        JP      SEN1            ;GO MOVE NEXT ONE
SEN2    CALL    DRIVE           ;STUFF DRIVE # IN FCB
        CALL    CLRHY           ;CLEAR OUT HI-MEM
        LD      A,195           ;RESET JUMP
        LD      (400CH),A       ;IN LO-MEM
        CALL    MOVDN           ;RESTORE DOS TO LO-ME!?
        CALL    OPEN1           ;OPEN AN EXISTING FILE
        CALL    MULPLY          ;CALCULATE # FILE BYTES
        CALL    READ            ;READ THEM FROM DISK
        CALL    CLOSE           ;THEN CLOSE THE FILE
        LD      SP,29758        ;RESET STACK POINTER
        CALL    SETUP           ;INITIALIZE PGM POINTERS
        CALL    CLRLO           ;CLEAR OUT DOS LO-MEM
        CALL    MOVHI2          ;MOVE FILE HI TO LO-MEM
        CALL    CALFRM          ;CALCULATE FRAMES/PACK
        CALL    CLRHY           ;CLEAR OUT HI-MEM
        CALL    CLS             ;C L E A R   V I D E O
        JP      PACK            ;SEND MULTI-FRAME/PACKS
DRIVE   LD      A,':'           ;DRIVE # SEPARATOR
        LD      (DE),A          ;STUFF INTO FCB
        LD      (BC),A          ;NOT USED NOW
        INC     DE              ;NEXT FCB LOCATION
        INC     BC              ;NOT USED NOW
        LD      A,'1'           ;DISK NUMBER
        LD      (DE),A          ;STUFF INTO FCB
        LD      (BC),A          ;NOT USED NOW
        INC     DE              ;NEXT FCB LOCATION
        INC     BC              ;NOT USED NOW
        LD      A,13            ;FILE NAME DELIMITER
        LD      (DE),A          ;STUFF INTO FCB
        LD      (BC),A          ;NOT USED NOW
        RET                     ;RETURN WHENCE U CAME +1
SAVEIT  DEFW    0               ;FILE NAME BEGIN STASH
NAM%    DEFM    'INPUT FILE NAME '
```

```
04820          DEFB    0                  ;DELIMITER
04830          ORG     0C700H             ;AUTO SAVE DISK FILE
04840 SAVE1    XOR     A                  ;ZERO OUT SAVE POINTER
04850          LD      (SIGN9),A          ;AT BEGIN RECEIVE MODE
04860          CALL    CLS                ;C L E A R   V I D E O
04870          LD      A.195              ;RESTORE JUMP
04880          LD      (400CH),A          ;TO LOW-MEM
04890          CALL    MOVDN              ;RESTORE DOS TO LO-MEM
04900          LD      HL,(SAVEIT)        ;BEGIN FILE NAME LOCATION
04910          LD      DE,FCB             ;FILE CTRL BLOCK ADDRESS
04920          LD      BC,NAME1           ;FILE NAME TEMP. STASH
04930 SAV1     LD      A,(HL)             ;FILE NAME BYTE
04940          CP      13                 ;FILE NAME DELIMITER
04950          JP      Z,SAV2             ;GOTO SAV2 IF DONE
04960          LD      (DE),A             ;NAME BYTE TO FCB
04970          LD      (BC),A             ;TEMPORARY STASH
04980          INC     HL                 ;NEXT NAME BYTE LOCATION
04990          INC     DE                 ;NEXT FCB LOCATION
05000          INC     BC                 ;NEXT STASH LOCATION
05010          JP      SAV1               ;CONTINUE MOVING NAME
05020 SAV2     CALL    DRIVE              ;LOAD DISK DRIVE NUMBER
05030          CALL    OPEN2              ;OPEN NEW FILE ONLY
05040          LD      SP,29758           ; RESET STACK POINTER
05050          CALL    SETUP              ;INITIALIZE PGM POINTERS
05060          CALL    CLRLO              ;CLEAR OUT DOS LO-MEM
05070          JP      SENDIT             ;SEND 'GO AHEAD' MESSAGE
05080 OPEN2    LD      HL,BUFFER          ;DISK I/O WORKING AREA
05090          LD      DE,FCB             ;FILE CTRL BLOCK ADDRESS
05100          LD      B,0                ;256 BYTES PER RECORD
05110          LD      C,10H              ;FILE TYPE DOUBTFUL
05120          CALL    4424H              ;OPEN AN EXISTING FILE
05130          JP      NZ,OPEN4           ;NZ = IT DOES NOT EXIST
05140          LD      SP,29758           ;NO DUPLICATION ALLOWED
05150          CALL    SETUP              ;SO RESET SP & INTXALIZE
05160          LD      IY,MS2B            ;NAME ALREADY USED MSG
05170          JP      SEND3Z             ;SEND 'TRY ANOTHER' MSG
05180 OPEN3    LD      HL,BUFFER          ;DISK I/O WORKING AREA
05190          LD      DE,FC%             ;FILE CTRL BLOCK ADDDRESS
05200          LD      B,0                ;256 BYTES PER RECORD
05210          LD      C,10H              ;FILE TYPE DOUBTFUL
05220 OPEN4    CALL    4420H              ;OPEN A NEW DISK FILE
05230          RET                        ;RETURN WHENCE U CAME +1
05240 SENDIT   CALL    CLRHY              ;CLEAR HI-MEM
05250          LD      A,1                ;SET
05260          LD      (AUTSAV),A         ;AUTO SAVE FILE POINTER
05270          LD      IY,MS3A            ;'GO AHEAD' MESSAGE
05280          JP      SEND3Z             ;SENT VIA SEND3
05290          ORG     0C780H             ;AUTO SAVE HI-MEM ONLY
05300 SAV3     LD      A,195              ;FOR FILES > 12K BYTES
05310          LD      (400CH),A          ;LENGTH 'IF' A DISCONNECT
05320          CALL    MOVDN              ,= ALL DONE IS RECEIVED.
05330          CALL    SETEND             ;STUFF 3 128 DELIMITERS
05340          CALL    SETFCB             ;MOVE FILE NAME INTO FCB
05350          CALL    OPEN1              ;OPEN AN EXISTING FILE
05360          CALL    4448H              ;POSITION END OF FILE
05370          CALL    DUMP               ;WRITE MEM TO DISK
05380          CALL    CLOSE              ;CLOSE THE DISK FILE
05390          LD      SP,29758           ; RESET STACK POINTER
05400          CALL    SETUP              ;INITIALIZE PGM POINTERS
05410          CALL    CLRLO              ;CLEAR OUT DOS LO-MEM
05420          CALL    CLRHY              ;CLEAR HI-MEM
05430          JP      CONREQ             ;RE-CONNECT TO STATION
05440 SETEND   LD      HL,(BEFORE)        ;END HI-MEM INFO FIELDS
05450          LD      A,128              ;STUFF
05460          LD      (HL),A             ;3 EACH
05470          INC     HL                 ;END OF MESSAGE
05480          LD      (HL),A             ;DELIMITERS
05490          INC     HL                 ;AT
05500          LD      (HL),A             ;THE
05510          INC     HL                 ;VERY END.
05520          LD      (SOFAR+1),HL       ;SAVE END MEM LOCATION
05530          RET                        ;RETURN WHENCE U CAME +1
05540 HOWFAR   LD      HL,53248           ;CALCULATE BYTES TO SAVE
05550 FAR1     INC     HL                 ;TO DISK FILE
05560          LD      A,(HL)             ;LOOK
05570          CP      128                ;FOR
05580          JP      NZ,FAR1            ;THREE
05590          INC     HL                 ;EACH
05600          LD      A,(HL)             ;DECIMAL
05610          CP      128                ;128
05620          JP      NZ,FAR1            ;END
05630          INC     HL                 ;OF
05640          LD      A,(HL)             ;MESSAGE
05650          CP      128                ;DELIMITERS
05660          JP      NZ,FAR1            ;IN A
05670          INC     HL                 ;ROW.
05680          LD      (SOFAR+1),HL       ;SAVE LOCATION IN SOFAR
05690          RET                        ;RETURN WHENCE U CAME +1
05700 DUMP     LD      HL,53248           ;BEGIN HI-MEM INFO FIELDS
05710          LD      DE,FCB             ;FILE CTRL BLOCK ADDRESS
05720 DUM1     LD      A,(HL)             ;BYTE TO SAVE ON DISK
05730          PUSH    HL                 ;SAVE BYTE MEM LOCATION
05740          CALL    1BH                ;WRITE BYTE TO DISK
05750          POP     HL                 ;RESTORE BYTE LOCATION
05760          JP      NZ,ERROR           ;Z FLAG SET IF ERROR
05770          INC     HL                 ;NEXT BYTE MEM LOCATION
05780          PUSH    HL                 ;SAVE IT IN STACK
05790          PUSH    DE                 ;SAVE FCB POINTER
05800 SOFAR    LD      DE,65535           ;LAST MEM BYTE LOCATION
05810          OR      A                  ;CLEAR CARRY FLAG
05820          SBC     HL,DE              ;SUBTRACT DE FROM HL
05830          POP     DE                 ;RESTORE FCB POINTER
05840          POP     HL                 ;AND NEXT MEM LOCATION
05850          RET     Z                  ;RETURN IF ALL DONE
05860          JP      DUM1               ;GO DUMP NEXT ONE TO DISK
05870 NAM1     DEFM    'REMEMBER 128 DELIMITERS ? HIT BREAK TO ES
CAPE ELSE <ENTER>'
05880          DEFB    0                  ;DELIMITER
05890 SETFCB   LD      HL,NAME1           ;FILE NAME TEMP. STASH
05900          LD      DE,FCB             ;FILE CTRL BLOCK ADDRESS
05910 ST1      LD      A,(HL)             ;MOVE FILE NAME+DELIMITER
05920          LD      (DE),A             ;TO FILE CTRL BLOCK
05930          CP      13                 ;DELIMITER ?
05940          RET     Z                  ;IF SO, ALL DONE
05950          INC     HL                 ;NEXT NAME LOCATION
05960          INC     DE                 ;NEXT FCB LOCATION
05970          JP      ST1                ;MOVE NEXT ONE
05980          ORG     0C840H             ;MANUAL SAVE DISK FILE
```

```
SVFILE  CALL    INPNAM          ;REMINDER+INPUT FILE NAME
        CALL    HOWFAR          ;CALCULATE BYTES TO SAVE
        LD      A,195           ;RESTORE JUMP
        LD      (400CH),A       ;TO LO-MEM
        CALL    MOVDN           ;RESTORE DOS TO LO-MEM
        CALL    OPEN3           ;OPEN A NEW FILE
        CALL    DUMP            ;DUMP HI-MEM TO DISK
        CALL    CLOSE           ;CLOSE THE DISK FILE
        LD      SP,29758        ;RESET STACK POINTER
        CALL    SETUP           ;INITIALIZE PGM POINTERS
        CALL    CLRLO           ;CLEAR DOS OUT LO-MEM
        JP      MENU            ;MAIN MENU FOR INSTRUCTS
MS2C    DEFB    13              ;CARRIAGE RETURN
        DEFB    10              ;LINE FEED
        DEFM    'ERROR # '      ;ERROR MESSAGE
MS2A    DEFM    '     <READY> ' ;ERROR # + <READY> MSG
        DEFB    13              ;CARRIAGE RETURN
        DEFB    10              ;LINE FEED
        DEFB    128             ;ONLY 2 DELIMITERS NEEDED
        DEFB    128             ;FOR SEND3 PROCESSING
        ORG     0C880H          ;MANUAL LOAD FILE TO MEM
LDFILE  CALL    INNAME          ;INPUT FILE NAME
        CALL    CLRHY           ;CLEAR HI-MEM
        LD      A,195           ;RESTORE JUMP
        LD      (400CH),A       ;TO LO-MEM
        CALL    MOVDN           ;RESTORE DOS LO-MEM
        CALL    OPEN1           ;OPEN EXISTING DISK FILE
        CALL    MULPLY          ;CALCULATE FILE LENGTH
        CALL    READ            ;LOAD FILE TO HI-MEM
        LD      (HIHL),HL       ;SAVE HI-MEM END LOCATION
        CALL    CLOSE           ;CLOSE DISK FILE
        LD      SP,29758        ;RESET STACK POINTER
        CALL    SETUP           ;INITIALIZE PGM POINTERS
        CALL    CLRLO           ;CLEAR OUT DOS LO-MEM
        CALL    BAKUP           ;CHECK TOO LONG LOADED ?
        JP      MENU            ;MENU FOR INSTRUCTIONS
HIHL    DEFW    0               ;END HI-MEM FILE LOCATION
BAKUP   LD      HL,(HIHL)       ;IF THE FILE LENGTH JUST
BAK1    DEC     HL              ;LOADED
        LD      A,(HL)          ;HAD RECORDS
        CP      28              ;LESS THAN
        JP      Z,BAK1          ; 256 BYTES
        CP      128             ;LONG,
        JP      Z,TESAGN        ;THEN
        INC     HL              ;OBVIOUSLY
        LD      (BEFORE),HL     ;THIS
        RET                     ;SIMPLE
TESAGN  DEC     HL              ;SUBROUTINE
        LD      A,(HL)          ;WILL
        CP      128             ;LOAD
        JP      NZ,BAK1         ;FAR
        DEC     HL              ;TOO
        LD      A,(HL)          ;MANY
        CP      128             ;BYTES FROM DISK TO MEM.
        JP      NZ,BAK1         ;BAKUP'S JOB IS TO TRY &
        LD      (BEFORE),HL     ;CORRECT THIS SITUATION.
        RET                     ;RETURN WHENCE U CAME +1
        ORG     0C900H          ;AFTER SAVE IN AUTO MODE
SAVMSG  XOR     A               ;ZERO OUT
        LD      (AUTSAV),A      ;AUTO SAVE POINTER
        LD      IY,MS4A         ;HAS BEEN SAVED MESSAGE
        JP      SEND3Z          ;TRANSMIT VIA SEND3
MS4A    DEFB    13              ;CARRIAGE RETURN
        DEFB    10              ;LINE FEED
        DEFM    'YOUR FILE HAS BEEN AUTOMATICALLY SAVED ON DISK.  IF YOU WISH'
        DEFB    13              ;CARRIAGE RETURN
        DEFB    10              ;LINE FEED
        DEFM    'TO TEST IT TRY THE SEND (FILE NAME) COMMAND.   <READY> '
        DEFB    13              ;CARRIAGE RETURN
        DEFB    10              ;LINE FEED
        DEFB    128             ;SEND 3 ONLY REQUIRES TWO
        DEFB    128             ;128 DELIMITERS
READY   LD      HL,MS5A         ;<READY> MESSAGE
        LD      BC,14           ;BYTES TO MOVE
        LDIR                    ;MOVE IT INTO 'DE'
        RET                     ;RETURN WHENCE U CAME +1
MS5A    DEFB    13              ;CARRIAGE RETURN
        DEFB    10              ;LINE FEED
        DEFM    '<READY>'       ;MESSAGE
        DEFB    13              ;CARRIAGE RETURN
        DEFB    10              ;LINE FEED
        DEFB    128             ;3 DELIMITERS
        DEFB    128             ;ARE NEEDED
        DEFB    128             ;BY THE MULTI-FRAME
        RET                     ;XMIT LO-MEM SUBROUTINE
MS1A    DEFM    'DIR :1'        ;DISK DRIVE NUMBER USED
        DEFB    13              ;USED.  CHANGE TO SUIT
        DEFB    10              ;YOUR FANCY.
MS2B    DEFB    13              ;CARRIAGE RETURN
        DEFB    10              ;LINE FEED
        DEFM    'FILE NAME EXTANT - TRY ANOTHER ONE. '
        DEFB    13              ;CARRIAGE RETURN
        DEFB    10              ;LINE FEED
        DEFB    128             ;DELIMITER
        DEFB    128             ;DELIMITER
NAME1   DEFS    13              ;13 BYTE FILE NAME STASH
MS3A    DEFB    73              ;CARRIAGE RETURN
        DEFB    10              ;LINE FEED
        DEFM    'SEND DISK FILE & WHEN DONE A DISCONNECT TO CLOSE FILE. '
        DEFB    13              ;CARRIAGE RETURN
        DEFB    10              ;LINE FEED
        DEFB    128             ;DELIMITER
        DEFB    128             ;DELIMITER
        ORG     0CA40H          ;SAVE VERY LONG FILES
SAV4    LD      A,195           ;AFTER A 'WAIT' RNR HAS
        LD      (400CH),A       ;HAS BEEN TRANSMITTED.
        CALL    MOVDN           ;RESTORE DOS TO LO-MEM
        LD      HL,(BEFORE)     ;END HI-MEM INFO FIELDS
        LD      (SOFAR+1),HL    ;LOAD INTO SOFAR
        CALL    SETFCB          ;SET FILE CTRL BLOCK
        CALL    OPEN1           ;OPEN AN EXISTING FILE
        CALL    4448H           ;POSITION TO END OF FILE
        CALL    DUMP            ;AND DUMP DATA TO DISK
        CALL    CLOSE           ;CLOSE DISK FILE
        LD      SP,29758        ;RESET STACK POINTER
```

```
07140         CALL    SETUP          ;INITIALIZE PGM POINTERS
07150         CALL    CLRHY          ;CLEAR HI-MEM
07160         LD      IY,MS6A        ;WAIT/RNR CLEARED MSG
07170         JP      SEND3Z         ;CONTINUE SENDING MESSAGE
07180 MS6A    DEFB    13             ;CARRIAGE RETURN
07190         DEFB    10             ;LINE FEED
07200         DEFM    '<RNR CLEARED - CONTINUE SENDING>'
07210         DEFB    13             ;CARRIAGE RETURN
07220         DEFB    10             ;LINE FEED
07230         DEFB    128            ;DELIMITER
07240         DEFB    128            ;DELIMITER
07250 VYLONG  LD      (NUMSEC),A     ;LOAD # OF' DISK SECTORS
07260         LD      DE,0           ;ZERO OUT COUNTER
07270         LD      B,47           ;MAX SECTORS IN HI-MEM
07280 VY1     SUB     B              ;SUBTRACT FROM TOTAL
07290         JP      Z,DON2         ;Z FLAG SET = ALL DONE
07300         JP      C,DON1         ;C FLAG SET = ALL DONE
07310         INC     D              ;SEGMENT COUNTER
07320         JP      VY1            ;TRY AGAIN
07330 DON1    INC     D              ;INCREMENT SEGMENT COUNT
07340 DON2    LD      A,D            ;SWAP INTO 'A'
07350         LD      (SEGNUM),A     ;SAVE IN NUMBER SEGMENTS
07360         DEC     A              ;DECREMENT SEGMENT COUNT
07370         LD      E,A            ;SWAP INTO 'E'
07380         XOR     A              ;ZERO OUT 'A'
07390         LD      B,47           ;MAX RECORDS PER SEGMENT
07400 DON3    ADD     A,B            ;ADD 'EM UP AGAIN
07410         DEC     E              ;MINUS ONE FROM COUNTER
07420         JP      NZ,DON3        ;NZ THEN DO IT AGAIN
07430         LD      B,A            ;THEN SWAP INTO 'B'
07440         LD      A,(NUMSEC)     ;NUMBER SECTORS REMAINING
07450         SUB     B              ;SUBTRACT COUNTER
07460         DEC     A              ;MINUS ONE
07470         LD      (NUMSEC),A     ;SAVE IT IN NUMBER SECTOR
07480         LD      A,1            ;ONLY 1
07490         LD      (SECTOR),A     ;SAVE IT IN SECTOR
07500         LD      HL,12032       ;MAX SEGMENT LENGTH
07510         JP      MUL3           ;AND GO SET LBYTCS TO MAX
07520 NUMSEC  DEFB    0              ;NUMBER SECTORS STASH
07530 SECTOR  DEFB    0              ;SINGLE SECTOR STASH
07540 NAM1A   DEFM    'INPUT BEGINNING MEM ADDRESS (53248 NOMINA
L) '
07550         DEFB    0              ;DELIMITER
07560         ORG     OCBOOH         ;ONLY USED FOR SENDING
07570 SEND2   CALL    CLRHY          ;DISK FILES GREATER THAN
07580         LD      A,195          ;12K BYTES - 65K MAXIMUM.
07590         LD      (400CH),A      ;USED AFTER 1ST 12K SENT.
07600         CALL    MOVDN          ;RESTORE DOS TO LO-MEM
07610         CALL    OPEN1          ;OPEN AN EXISTING FILE
07620         CALL    ANYMOR         ;CALCULATE MORE TO SEND
07630         CALL    READ           ;READ FROM DISK TO HI-MEM
07640         CALL    CLOSE          ;CLOSE DISK FILE
07650         LD      SP,29758       ;RESET STACK POINTER
07660         CALL    SETUP          ;INITIALIZE PGM POINTERS
07670         CALL    CLRLO          ;CLEAR OUT DOS LO-MEM
07680         CALL    MOVHI2         ;MOVE HI-MEM TO LO-MEM
07690         CALL    CALFRM         ;CALCULATE FRAMES TO SEND
07700         CALL    CLRHY          ;CLEAR HI-MEM
07710         JP      PACK           ;SEND MULTI-FRAME PACKS
07720 ANYMOR  LD      A,(SECTOR)     ;SECTOR REMAINING
07730         LD      B,47           ;MAX SECTORS FOR MEM
07740         ADD     A,B            ;ADD 'EM UP
07750         LD      (SECTOR),A     ;AND SAVE IN SECTOR
07760         LD      A,(SEGNUM)     ;NUMBER OF SEGMENTS
07770         CP      1              ;LAST ONE ?
07780         JP      Z,MUL4         ;THEN CALCULATE LENGTH
07790         LD      HL,12032       ;MAX BYTES PER SEGMENT
07800         CALL    MUL3           ;SET LBYTES LENGTH
07810         LD      DE,FCB         ;FILE CTRL BLOCK ADDRESS
07820         LD      A,(SECTOR)     ;NUMBER OF DISK SECTORS
07830         LD      B,0            ;256 BYTES PER RECORD
07840         LD      C,A            ; SWAP SECTORS INTO 'C'
07850         CALL    4442H          ;POSITION TO DISK SECTOR
07860         RET                    ;RETURN WHENCE U CAME +1
07870 MUL4    LD      DE,FCB         ;ONLY LAST FILE SEGMENT
07880         LD      A,(SECTOR)     ;LAST FILE SECTOR READ
07890         LD      B,0            ;256 BYTES PER RECORD
07900         LD      C,A            ; SWAP SECTORS INTO 'C'
07910         CALL    4442H          ;POSITION TO DISK SECTOR
07920         LD      A,(NUMSEC)     ;NUMBER SECTORS REMAINING
07930         CALL    MUL0           ;CALCULATE BYTES TO READ
07940         RET                    ;RETURN WHENCE U CAME +1
07950 ; - - - - - - - - - - - - - - - - - - - - - - - - - - - -
07960 ; END OF DISK I/O FOR AUTO MODE
```

- Figure 2 -

HELP Message Transmitted in Auto Mode

```
MJTHIS IS W4UCH CHAUTAUQUA LAKE, NY IN AUTOMATIC DISK I/O MODEMJ
   THERE ARE FOUR FUNDAMENTAL INSTRUCTIONS THAT THIS MODE WILLMJ
   AUTOMATICALLY RECOGNIZE AFTER YOU ARE CONNECTED. THEY ARE: MJ
1. A SINGLE INFO PACKET CONSISTING OF HELP TO CALL THIS       MJ
   PRESENT  SUBROUTINE.                                        MJ
2. A SINGLE INFO PACKET CONSISTING OF LIST TO CALL THE DISK   MJ
   DIRECTORY SUBROUTINE.                                       MJ
3. A SINGLE INFO PACKET CONSISTING OF SEND - SPACE - NAME OF  MJ
   DISK PROGRAM, AND CARRIAGE RETURN TO READ THE DISK PROGRAM.MJ
4. A SINGLE INFO PACKET CONSISTING OF SAVE - SPACE - NAME OF  MJ
   DISK PROGRAM, AND CARRIAGE RETURN TO WRITE DISK PROGRAM.   MJ
   THE PROGRAM WILL THEN OPEN A DISK FILE AND RESPOND WITH THEMJ
   MESSAGE, 'SEND FILE.' GO AHEAD AND SEND THE FILE.  IF THE  MJ
   FILE IS LONGER THAN 10K OR SO BYTES IN LENGTH, THE PROGRAM MJ
   WILL SEND A 'WAIT' RNR WHILE IT SAVES THE DATA ON DISK, ANDMJ
   WHEN SAVED, WILL SEND AN AUTOMATIC 'CLEAR WAIT' RR WHEN    MJ
   READY FOR YOU TO RESUME SENDING THE FILE. THE MODEL 1 TRS80MJ
   DISK CAPACITY IS ABOUT 85K BYTES AND THE MODEL 3TRS80 DISKMJ
   CAPACITY IS ABOUT 170K BYTES.                               MJ
                                                               MJ
   SINGLE FRAME PACKETS MAY BE UP TO 2000 BYTES IN LENGTH AND MJ
   7 FRAME PACKETS MAY HAVE INFO FIELDS OF UP TO 256 BYTES.   MJ
   JUST HOLD TOTAL PACKET LENGTH TO 2000 BYTES OR LESS.       MJ
                                                               MJ
   WHEN YOU HAVE FINISHED SENDING THE DATA YOU WISH SAVED ON  MJ
   DISK, SEND A DISCONNECT.  THIS TELLS THE PROGRAM TO FINISH MJ
   SAVING YOUR DATA ON DISK AND 'CLOSE' THE FILE. WHEN THIS ISMJ
   DONE, THE PROGRAM WILL SEND AN AUTOMATIC CONNECT REQUEST   MJ
   AND UPON RECEIVING THE CONNECT REQUEST ACKNOWLEDGE WILL    MJ
   TELL YOU THAT THE DATA WAS SAVED. SHOULD YOU WISH TO CHECK MJ
   IT, GO AHEAD AND DO SO WITH THE SEND - SPACE - FILE NAME - MJ
   COMMAND.                                                    MJ
                                                               MJ
SUGGEST YOU USE THE FOLLOWING CONVENTION FOR FILE NAMES:      MJ
A. FILE NAMES MAY BE UP TO 8 UPPERCASE ALPHANUMERIC CHARACTERSMJ
   AND 'MUST' BEGIN WITH AN ALPHABETIC CHARACTER.             MJ
B. OBJECT CODE FILES SHOULD END WITH A NUMERAL 1 OR /CMD.     MJ
C. SOURCE CODE FILES SHOULD END WITH A NUMERAL 2.             MJ
D. BASIC PROGRAM FILES SHOULD END WITH A NUMERAL 4.           MJ
E. PLAIN VANILLA ASCII FILES/MESSAGES SHOULD BE ALL CAPITALS. MJ
F. ELECTRIC PENCIL FILES SHOULD END WITH /PCL.                MJ
                                                               MJ
THERE ARE 'NO' PROTECTED FILES ON THE DISKFILE DISK. THIS IS MJ
INTENTIONAL. YOU MAY 'READ' ANY EXISTING FILE ON THE DISK.    MJ
WHEN ATTEMPTING TO TO WRITE TO AN EXISTING FILE YOU WILL      MJ
RECEIVE THE MESSAGE 'FILE ALREADY EXTANT - TRY ANOTHER ONE.' MJ
IN THIS CASE, JUST GIVE THE FILE YOU WISH SAVED ANOTHER NAME MJ
AND TRY AGAIN.                                                 MJ
                                                               MJ
SHOULD YOU INADVERTENTLY BUGGER-UP SENDING A GIVEN FILE, FIRSTMJ
CLOSE THE FILE BY SENDING A DISCONNECT. THE PROGRAM WILL AUTO-MJ
MATICALLY RECONNECT TO YOU. NOW RENAME THE FILE AND TRY AGAIN.MJ
DO NOT FORGET THE DISCONNECT TO CLOSE THE FILE. IF SO, IT MAY MJ
BUGGER-UP THE PROGRAM FOR YOUR FELLOW AMATEURS WHO MAY WISH TOMJ
USE THE AUTO MODE LATER.                                       MJ
                                                               MJ
REMINDER: CAPITAL LETTERS ARE REQUIRED FOR ALL COMMANDS AND   MJ
FILE NAMES AS SOME USERS MAY NOT HAVE THE LOWERCASE CAPABILITYMJ
AND WOULD NOT BE ABLE TO READ THE FILE NAMES.        MJ<READY>MJ
```

Note:
The capital M = ASCII 13 carriage return and the capital J = ASCII 10 line feed.

KEYBOARD INPUT MESSAGE WITH AUTOMATIC SWITCHING IN CONNNECTED MODE AND RETURN. FOR PACKET RADIO USING SOFTWARE APPROACH

Robert M. Richardson, W4UCH
22 North Lake Drive
Chautauqua Lake, N.Y. 14722

ABSTRACT:

A number of options open to the programmer are discussed including manual switching, using interrupts, and the psuedo interrupt mode. The objective is to emulate as closely as possible the packet radio hardware approach (VADCG and TAPR TNCs) which utilize their own microprocessor, ROM, and RAM plus a separate host microcomputer. Obviously, a single microcomputer using the software approach does not have the capability of TWO microcomputers (the VADCG and TAPR TNCs are dedicated packet microcomputers), but with judicious programming this problem may be finessed.

INTRODUCTION:

There are many options open to the programmer who is writing subroutines for keyboard message input. A number of approaches that could be used in the connected mode are:

1. While keyboard inputting messages, ignore ALL incoming packets till the operator chooses to switch to receive mode. We will label this approach, IGNORE TILL READY. It is the approach used by the author's Volume 2 - AX.25 Protocol.

2. Use the Model I/III TRS-80's Z-80 microcomputer interrupt mode 1 to switch back and forth between keyboard input and transmit/receive mode data processing. This may be accomplished by splitting both the transmit AND receive 'bit' time delay countdowns between each bit processed into quadrants and using the quadrants for sequentially doing the keyboard processing; i.e, scanning the keyboard psuedo memory locations, converting the keyboard input to ASCII, and stashing the ASCII byte into memory, sequentially during each quadrant's idle countdown time. We will label this approach, INTERRUPT MODE 1. It is a rather fascinating challenge, but not all that difficult. Its major detriments are the amount of memory required and the considerable care that must be exercised to avoid disturbing the software digital phase locked loop while in receive mode or the 'bit' countdown timing in transmit mode.

3. There is yet another approach. While keyboard inputting a packet message to be transmitted in connected mode, have the program test the input from the receiver (via the EXAR 2211 AFSK demodulator and port zero) every millisecond or so and IF a valid change occurs, then automatically switch the program to receive mode where the incoming packet is decoded. If not for your station, then automatically switch back to keyboard input mode. If for your station and it was received correctly, then acknowledge the packet and then automatically switch back to keyboard input mode. This is the approach this paper will discuss with a few enhancements. We will call this the PSUEDO INTERRUPT approach since the change in input in essence brings about an interrupt service subroutine.

GENERAL:

This software approach uses the 1024 byte page of memory beginning at 28672 in memory for keyboard inputting short single frame packets. Long multi-frame packets, up to 12K, are input to low memory beginning at 17408 decimal. During the balance of this discussion we will presume you are already connected to another station. As such, you would press V from the main menu (Figure 1) which takes the program to the edit/modify mode and displays a 1024 byte page of zeros beginning at 28672 decimal in memory.

To activate the modify mode, press M to light the blinking rectangular cursor in the upper left hand corner of the video display. Most any key pressed will insert its ASCII value into the video display AND the corresponding memory location except for the arrow keys which move the cursor on the page rather quickly within the page boundaries, the shift zero keys which insert sequential decimal 128 end of message delimiters;, and the shift @ keys which display the decimal value beneath the cursor.

Figure 2 is the commented source code for that segment of the edit/modify subroutine that performs the blinking cursor operation in modify mode, plus the test for a valid change from the receiver. By valid change we mean:

A. That the EXAR 2211 has noted a change from a mark to space, or vice versa.

B. And that the EXAR 2211 data carrier detect (DCD) has not dropped.

Blink A starting in line 5760 stashes the value in video memory (IX) beneath the blinking cursor in the memory location noted by the label HOLDIT and then loads the same video location with the rectangular cursor character = 143 decimal. TEZRCV in line 5860 is then CALLed.

Lines 5860 - 5980:
First save registers BC, DE, HL, IX, and IY in the stack and then start a few millisecond countdown. Note that lines 5880 and 5910 sample the EXAR 2211 output via port zero, and if any change occurs, then line 5930 jumps off to LOOK in line 5990. If no change occurs during the countdown, line 6100 restores the saved registers from the stack and then returns whence called + 1.

Blink B starting in line 5820 simply replaces the stored character back onto video, CALLS TEZRCV, and then returns. Blink A and B are alternately called frequently enough to comfortably display the blink effect about 30 times a second.

Lines 5990 - 6110:
First test to see if the operator is in the connected mode = CONEK1 set to 1. If not, then line 6010 jumps back to the TEZRCV countdown. If connected, then WATE is tested to see if the operator had previously transmitted a WAIT (RNR) command, and if so, then jumps back to the TEZRCV countdown. The EXAR 2211 DCD is then tested and if dropped (no mark or space tone present), then jumps back to the TEZRCV countdown. If it gets this far, to line 6080, then SIGN10 is set = come back to the modify mode, the receive mode video display restored in 6100, and then the programs jumps off to receive mode to process the incoming packet. When done with the incoming packet, SIGN10 at the beginning of the receive mode subroutine sends the program off to line 6120.

Lines 6120 - 6200:
First zero out SIGN10, then save the receive mode video display in memory, restore the modify mode video display exactly the way it was before, restore all the modify mode registers, and lastly return to wherever the program was in the modify mode.

OPERATION:

When connected to another packeteer on an otherwise quiet non-repeater 2 meter frequency, this approach works perfectly. When on a packet digi-peater frequency it works very well as long as activity is low; i.e., not more than 2 pairs of stations on frequency who are NOT sending long files back and forth.

IF on a busy packet digi-peater frequency, one has no choice but to send the other station a WAIT (RNR) command from the main menu by pressing 3, type in the keyboard input message, and then send it from the main menu by pressing V. Your info packet clears your previous WAIT (RNR) command at the other end of the circuit.

Another approach we tried was having the program send an automatic WAIT (RNR) command whenever you entered the modify mode when connected. This of course worked quite well, but seemed a rather needless transmission on a quiet channel, so was removed and left for the operator to perform manually when desired. Press 3 from the main menu.

CONCLUSION:

This simple subroutine hopefully removes the last objection to our software approach by dyed in the wool hardcore hardware approach packeteers who get their jollies by pointing out subliminal differences between the 2 approaches. It is fully implemented in the author's 'Advanced Vancouver Protocol - Software Approach* program.

It may easily be implemented in the author's 'AX.25 Protocol - Software Approach' program if desired. During BETA testing of the Volume 2 - AX.25 Software Approach program, about 1/2 of the BETA testers wanted this function, while the other 1/2 saw no need for it as the program allows the operator to switch to receive mode manually if desired. Since the vote was a draw, we chose to omit it from Vol. 2 - AX.25.

If you would like a disk for the 'Advanced Vancouver Protocol - Software Approach' source & object programs, uncommented, with no instructions whatsoever, you are on your own, then send $29 for the disk (specify Model I or Model III TRS-80) to:

Richcraft Engineering Ltd.
#1 Wahmeda Industrial Park
Chautauqua, New York 14722

Alternatively, call Richcraft at (716) 753-2654 on weekdays during business hours for COD shipment in the U.S.

This program includes automatic disk file access in the AUTO mode by the station connected to your station, as mentioned elsewhere in these conference proceedings.

FIGURE 1

VOLUME 3 MAIN MENU

```
                    ENTER OPTION DESIRED ? _

CHANGE ADDRESSEE CALL       = A    VE3NEC CONNECT REQUEST CQ = B
NOW CONNECTED TOGGLE        = C    VE3NEC DISCONNECT REQUEST = D
SEND PACKETS FROM LO-MEM    = E    VE3NEC CONNECT ACKNOWLEDGE = F
INPUT' FRAMES/PACKET LO-MEM = G    THIS IS VANCOUVER PROTOCOL = H
BACKOFF DELAY TOGGLE ON     = I    AUTO CONNECT TOGGLE ON     = J
NOW IN UPPER CASE MODIFY    = K    W2EUP - GIL BOELKE MESSAGE = L
DISPLAY/EDIT MEMORY PAGE    = M    SET INFO FIELD LOMEM PACKS = N
NOW FORMAT VIDEO TOGGLE     = O    QUICK BROWN FOX MESSAGE    = P
TRANSMIT EXTERNALLY ONLY    = Q    SET OPENING FLAG LENGTH    = R
TRANSMIT TO HI-MEM ONLY     = S    INPUT/XMIT NORMAL INFO = V & T
CLEAR NON-PGM MEM 17K-62K   = U    INPUT/XMIT ALL STATION = V & W
ABORT LOW-MEM PAK SEQUENCE  = X    SET RE-TRY IN CONNECT MODE = Y
SHIFT MENU                  = 1    MOVE HI-MEM TO LOW-MEM     = 2
SEND WAIT REQUEST (RNR)     = 3    SEND CLEAR WAIT (RR)       = 4
not shown:                         not shown:
HI TO LO-MEM INSERT CR/LF <---     MOVE LOW TO HIGH MEMORY    --->
```

VOLUME 3 SHIFT MENU

```
                    SHIFT MENU ? _

XMIT 40960 UP CONTINUOUSLY = A     BOOT DOS READY               = B
LOAD HI-MEM ASCII UUUUUUU = C      LOAD HI-MEM LOGIC 1111111 = D
EDIT/MODIFY INSTRUCTIONS  = E      CHANGE RECEIVE DPLL BASE # = F
LOG ON VE3MHZ REPFSTER     = G     LOG OFF VE3MHZ REPEATER      = H
SEND MORSE I.D.            = I     SEND SEQUENTIAL ACKS         = J
CAUTION ** RESTORE DOS **  = K     DISPLAY LOW MEMORY @ 17408 = L
DISPLAY RECV PACKS @ 53248 = M     RESTORE PROGRAM POINTERS   = N
DISPLAY CALL/ADDRESS LIST  = O     MOVE PROGRAM TO LOW MEMORY = P
SAVE HI-MEM ON DISK        = Q     LOAD DISK FILE TO HI-MEM     = R
TRANSMIT BAUD RATE SELECT  = S     SEND DISK :1 DIRECTORY       = T
CLEAR HI-MEMORY 53248 +    = U     RECEIVE VANCOUVER PROTOCOL = V
RECEIVE AX. 25 NOT CONNECT = W     SEND MORSE FROM KEYBOARD     = X
NORMAL DISPLAY - NOT DPLL  = Y     DISPLAY DPLL LAST QUADRANT = Z
      NOTE: SPACE BAR IN RECEIVE MODE = RESEND LAST PACK
```

FIGURE 2

```
;
;PSUEDO INTERRUPT AUTOMATIC SWITCH TO RECEIVE SUBROUTINE

BLINKA  LD    A,(IX)          ;VIDEO MEM BYTE VALUE
        LD    (HOLDIT), A     ;SAVE IT IN HOLDIT
        LD    A, 143          ;RECTANGULAR CURSOR
        LD    (IX),A          ;DISPLAY IT ON VIDEO
        CALL  TEZRCV          ;TEST INPUT CHANGE
        RET                   ;RETURN WHENCE U CAME +1
BLINKB  LD    A,(HOLDIT)      ;PREVIOUS VIDEO CHARACTER
        LD    (IX),A          ;DISPLAY IT ON VIDEO
        CALL  TEZRCV          ;TEST INPUT CHANGE
        RET                   ;RETURN WHENCE U CAME +1
TEZRCV  CALL  SAVE            ;SAVE IY,IX,HL,DE,BC
        LD    BC, 100         ;COUNTDOWN VALUE
TZ1     IN    A,(O)           ;TEST PORT ZERO
        LD    D,A             ;STASH IT IN 'D' REGISTER
        DEC   BC              ;DECREMENT COUNTDOWN
        IN    A,(O)           ;TEST PORT ZERO AGAIN
        CP    D               ;ANY CHANGE ?
        JP    NZ,LOOK         ;IF SO, TEST DCD
        LD    A,B             ;TEST COUNTDOWN
        OR    C               ;FOR ZERO
        JP    NZ,TZ1          ;IF NOT, CONTINUE COUNT
        CALL  RSTOR           ;RESTORE IY,IX,HL,DE,BC
        RET                   ;RETURN WHENCE U CAME +1
LOOK    LD    A,(CONEK1)      ;CONNECTED POINTER
        CP    1               ;1 = CONNECTED
        JP    NZ,TEZRCV+3     ;IF NOT CONNECTED, IGNORE
        LD    A,(WATE)        ;WAIT RNR POINTER
        CP    1               ;1 = WAIT PREVIOUSLY SENT
        JP    Z,TEZRCV+3      ;IF SO, THEN IGNORE
        IN    A,(O)           ;TEST INCOMING VIA PORT
        BIT   O,A             ;DATA CARRIER DETECT ?
        JP    Z,TEZRCV+3      ;IF NOT, THEN IGNORE
        LD    A,1             ;ELSE SET RETURN TO
        LD    (SIGN1O),A      ;MODIFY MODE POINTER
        CALL  RESRCV          ;RESTORE RECEIVE VIDEO
        JP    BEFOR1          ;PROCESS INCOMING PACKET
RESMOD  XOR   A               ;RESTORE MODIFY MODE
        LD    (SIGN1O),A      ;ZERO OUT MODIFY POINTER
        CALL  SAVRCV          ;SAVE RECEIVE VIDEO
        LD    HL,28672        ; BEGIN MODIFY MEMORY
        LD    DE,15360        ; BEGIN VIDEO MEMORY
        LD    BC,1024         ;FULL PAGE TO DISPLAY
        LDIR                  ; MOVE THEM TO VIDEO
        CALL  RSTOR           ;RESTORE MODIFY REGISTERS
        RET                   ;RETURN WHENCE U CAME +1
;NOTE:
;(WATE) IS RESET TO ZERO WHEN PACKET IS TRANSMITTED.
```

ADDING MULTIPLE REPEATER CAPABILITY TO PACKET RADIO USING THE SOFTWARE APPROACH AX.25 VOL.2

Robert M. Richardson, W4UCH
22 North Lake Drive
Chautauqua Lake, N.Y. 34722

ABSTRACT:

A brief assembly language subroutine to add multiple repeater call letter decoding within the address field of a received AX.25 frame is described. If the operator's call and SSID are included in the multiple repeater segment of the address field of the received frame, the SSID has been repeated bit is set for each frame, each frame re-CRC'ed, and the entire packet then re-transmitted (forwarded) automatically by the program. Also, a short subroutine to allow the operator to input multiple repeaters' call letters into the address field of a packet to be transmitted is mentioned.

INTRODUCTION:

Since implementation of the level/layer 3 of the AX.25 packet protocol is taking somewhat longer than expected, a number of enterprising and hardy souls have promulgated the interim concept of adding multiple repeater call letters to the AX.25 address field. Theoretically, if all the stations' whose calls are in the repeater segment of each frame's address field are 'on-the-air' at the same time, have antenna systems capable of receiving one station in the repeater segment of the address field AND transmitting to another station in the repeater segment of the address field, then this interim concept may work.

We are not suggesting that this concept is theoretically unsound, but wish to point out the difficulties of making it work reliably and effectively within the real life amateur radio community. There is no question that in the laboratory it will work perfectly every time. There is no question that within the same metropolitan area it will work perfectly some of the time, Nevertheless, it is a fun and games option, so we doff our collective hats to those intrepid packeteers who created this fascinating feature. So as not to be the weird kid on the block who said "the king has no clothes on at all," we too have implemented this interesting option.

In our software approach program we have allocated 2048 bytes normal and 4096 bytes maximum, of memory for unprocessed, converted, received 8 bit parallel bytes per packet. This memory allocation allows the storage and automatic forwarding of 7 frame packets with maximum info field length (256 bytes) with up to forty four (44) repeater calls with SSIDs, also included in the extended address field of each frame.

Using the software appproach it is just as easy to check the repeater segment of the address field of each received frame for the operator's call letters for up to 44 repeaters as it is for 1 repeater, so since the name of the game is multiple repeaters, let's do it.

MODIFYING AX. 25 SOFTWARE APPROACH PROGRAM FOR FORWARDING WITH MULTIPLE REPEATERS:

Is illustrated in Figure 1's source code. Line numbers are for volume 2 of 'Packet Radio Using The Software Approach - AX.25 Protocol.' The commented source code is largely self explanatory. The only lines changed or added are: 12460 & 12470, 12505 - 12507, 12750, and 13101 - 13124. The program logic and flow follows.

TEZFOR (test forward) in line 12400 is entered after the packet had been received and decoded in real-time, and each frame passed the CRC test.

Lines 12400 -12450:
Determine the location of the frame's control byte (end of address field + 1) and store it in (RCTL).

Lines 12460 - 12470:
Modify the CAL (call letters comparison) subroutine beginning in line 13040 so that line 13070's JP,NZ is to TEZNUM (test number of repeater calls in address field),

Lines 12480 - 12506:
Add 14 decimal to the frame's beginning address in memory, which is the beginning of the repeater calls, if any, and save it in BGNRPT. CALRPT (calculate number of repeaters) in line 13311 is then called.

Lines 13111 - 13122:
Simply subtract the beginning repeater memory location address from the frame's control byte address location and if zero (no repeater calls in frame), go on to TESADR (test address) to see if the packet is addressed to vou. If there are 1 or

more repeater calls in the address field, then the amount of memory used by the repeater call(s) is divided by 7 (call letters + SSID) and the number of repeater calls stashed in NUMRPT before returning to line 12507.

Lines 12507 - 12520:
Load HL with the first repeater's call letter first byte memory location, load DE with your call letter first byte memory location, and then call CAL in line 13040.

Lines 13040 - 13110:
Scan through the frame's extended address field searching for a match between your call letters and the call letters in the repeater segment of the frame's address field. If no match is found, then line 13115 jumps off to TESADR to see if the frame was addressed to vou and if so, then process it. If a match-is found, then line 13101 returns to line 12530.

Lines 12530 -12550:
First test the repeater's SSID against yours. The program assumes that your SSID byte's bit one is zero (if not, change it accordingly in line 12530). If not the same, line 12540 jumps off to TEZADR. If the same, then RECRC is called to set the 'has been repeated SSID bit' and re-CRC the frame.

Lines 12560 - 12630:
Test the P/F bit of the frame's control byte and if not set = more frames in this packet, jump off to process the next frame in line 12710. If the P/F bit is set = last frame of this packet, lines 12600 - 12610 set alternate DE with LENG1 (total length of packet + 1) and then jump off to REXIT to re-transmit (forward) the packet.

All this processing only requires a few milliseconds and is totally transparent to the operator except for the-<FORWARDING> message which is displayed on the receive mode video display.

MODIFYING AX. 25 SOFTWARE APRROACH TO TRANSMIT MULTI-REPEATER CALLS:

Is quite simple if only single frame packets are used. It seems to us that when using the multi-repeater function it would be wise to limit the packet to the single frame variety to keep the BERP (bit error rate probability) as low as possible. Further, 2 repeaters in the address field seems adequate for most all practical purposes.

If you wish to add the multi-frame packet capability when using multiple repeaters, the software approach gives you total freedom to do so. The only limitation in our software approach is the memory set aside for assembled packets ready to be transmitted = 2048 bytes. Therefore, the program without too much modification can accomodate maximum length info fields (256 bytes) with multiple repeaters:

7 frames per packet = 2 repeaters
6 frames per packet = 8 repeaters
5 frames per packet = 18 repeaters
4 frames per packet = 35 repeaters

If you wish to add the multi-frame AND multi-repeater transmit capability to our software approach, by all means do so and we wish you well.

CONCLUSION:

Having more than one repeater in the address field of an AX.25 frame is certainly a temporary and possibly useful expedient until level/layer 3 is implemented.

IF you would like a 35 track double sided disk for the Model I or single sided disk for the Model III TRS-80 with the multi-repeater capability in receive mode and up to two (2) repeaters input in transmit mode, then send $29 in US funds to:

Richcraft Engineering Ltd.
#1 Wahmeda Industrial Park
Chautauqua, New York 14722

A short single sheet of operating instructions is sent with the AX.25 disk outlining ONLY those changes to the operating instructions in Volume 2 of the software approach. Volume 2 is required for the balance of instructions to operate the program. The disk includes the PACK/CMD program, ASCII/CMD and MODIF1 object code programs, and ASCII2 and MODIF2 uncommented source code programs.

These modified programs also include the automatic switching from keyboard input message to receive mode function when connected, that is mentioned in another paper in these proceedings. The programs are very difficult to follow as it was necessary to move the real-time receive mode decoding subroutine from ASCII2 to the end of MODIF2 to allow the program to be assembled with a standard 2 pass editor & assembler in 48K of memory. Expert assembly language programmers should have little difficulty following these changes, so be forewarned as we do not plan to re-write volume 2 for these modest improvements.

IF you are the original purchaser of an earlier version of the Richcraft Ax.25 disk program and wish it updated, return the original disk and $10 to have it updated and returned to you postpaid.

- FIGURE 1 -

```
12400 TEZFOR  LD     HL,(BGINIT)    ;BEGIN FRAME IN MEMORY
12410         LD     A,(HL)         ;LOOK FOR THE FRAME'S
12420         INC    HL             ;CONTROL BYTE MEM ADDRESS
12430         BIT    0,A            ;AFTER THE
12440         JP     Z,TEZFOR+3     ;LAST SSID EYTE
12450         LD     (RCTL),HL      ;AND SAVE IT IN RCTL
12460         LD     HL,TESADR-1    ;CHANGE JP NZ ADDRESS
12470         LD     (CAL+6),HL     ;IN CALL LETTERS TEST
12480         LD     HL,(BGINIT)    ;BEGIN FRAME MEM LOCATION
12490         LD     DE,14          ;UR CALL+HIS CALL+SSID'S
12500         ADD    HL,DE          ;REPEATER ADDRESS IF ANY
12505         LD     (BGNRPT),HL    ;REPEATER BEGIN LOCATION
12506 .       CALL   CALRPT         ;CACULATE NO. REPEATERS
12507         LD     HL,(BGNRPT)    ;REPEATER BEGIN LOCATION
12510         LD     DE,FM          ;YOUR CALL LETTERS BEGIN
12520         CALL   CAL            ;COMPARE WITH REPEATER
12530         BIT    1,(HL)         ;TEST SSID FOR YOU ?
12540         JP     NZ,TEZADR      ;IF NOT, IGNORE IT
12550         CALL   RECRC          ;SET RPTR BIT+ RE-DO CRC
12560         LD     HL,(RCTL)      ;CONTROL BYTE LOCATION
12570         BIT    4,(HL)         ;P/F BIT SET = LAST ONE
12580         JP     Z,ADDIT        ;IF NOT, DO NEXT FRAME
12590         EXX                   ;SINCE LAST ONE THEN
12600         LD     DE,(LENG1)     ;SET ALTERNATE DE TO THE
12610         INC    DE             ;TOTAL PACKET
12620         EXX                   ;LENGTH + 1 FOR SEND7
12630         JP     REXIT          ;AND RE-TRANSMIT IT
12640 ADDIT   LD     A,(FRMNUM)     ;FRAMES PER PACK COUNTER
12650         INC    A              ;PLUS ONE
12660         LD     (FRMNUM),A     ;AND SAVE IT
12670         EXX                   ;SET ALTERNATE HL
12680         LD     ML,(LENG1)     ;FOR TOTAL PACK
12690         INC    HL             ;+ 1 FOR SEND7
12700         EXX                   ;RESTORE REG. REGISTERS
12710         JP     CONT           ;GO PROCESS NEXT FRAME
12720 FRMNUM  DEFB   0              ;FRAMES/PACK COUNTER
12730         POP    AF             ;ADJUST STACK FOR CALL
12740 TESADR  LD     HL,MODE2-1     ;RESET JP,NZ ADDRESS
12750         LD     (CAL+6),HL     ;IN CALL COMPARISON
12760         LD     HL,(BGINIT)    ;BEGIN FRAME MEM LOCATION
12770         LD     DE,FM          ;YOUR CALL LETTERS BEGIN
12780         CALL   CAL            ;COMPARE WITH YOUR CALL
12790         LD     A,(RPT)        ;VIA REPEATER POINTER
12800         CP     1              ;1 = ON 2 = OFF
12810         JP     Z,TES3         ;IF SO, TEST RPTR CALL
12820         LD     DE,7           ;NOT VIA REPEATER,
12830         ADD    HL,DE          ;SO TEST OTHER STATION'S
12840         BIT    0,(HL)         ;SSID BIT ZERO TO ENSURE
12850         JP     Z,MODE2        ;IT IS DIRECT OR IGNORE.
12860 TES2    LD     A,(SIGN3)      ;AUTO MODE POINTER
12870         CP     1              ;1 = ON 0 = OFF
12880         JP     Z,TESCTL       ;IF AUTO, TEST CONTROL
12890         LD     HL,(BGINIT)    ;ELSE TEST HIS
12900         LD     DE,7           ;CALL LETTERS
12910         ADD    HL,DE          ;AGAINST THE
12920         LD     DE,TO          ;CALL TO WHICH
12930         CALL   CAL            ;YOU ARE CONNECTED.
```

- FIGURE 1 CONTINUED -

```
12940         JP     TESCTL         ;OK, SO TEST CONTROL
12950 TES3    INC    DE             ;VIA REPEATER
12960         PUSH   DE             ;SO TEST
12970         LD     DE,8           ;REPEATER
12980         ADD    HL,DE          ;CALL
12990         POP    DE             ;LETTERS
13000         CALL   CAL            ;AND IF OK,
13010         BIT    7,(HL)         ;THE REPEATED SSID BIT.
13020         JP     Z,MODE2        ;NOT REPEATED, SO IGNORE
13030         JP     TES2           ;NOW TEST HIS CALL LTRS
13040 CAL     LD     BC,6           ;RPTR/CALL COMPARISON
13050         LD     A,(DE)         ;FRAME MID-MEM ADDRESS
13060         CP     (HL)           ;UR CALL LETTERS ADDRESS
13070         JP     NZ,MODE2-1     ;NOT SAME ? THEN EXIT
13080         INC    DE             ;NEXT MID-MEM ADDRESS
13090         INC    HL             ;NEXT COMPARISON ADDRESS
13100         DEC    C              ;-1 CALL LETTER COUNTER
13101         RET    Z              ;ALL MATCH, SO RETURN
13102         JP     CAL+3          ;GO TEST NEXT LETTER
13103 TEZNUM  LD     A,(NUMRPT)     ;NUMBER REPEATER CALLS
13104         DEC    A              ;LESS ONE
13105         LD     (NUMRPT),A     ;AND SAVE IT IN MEMORY
13106         JP     Z,TESADR-1     ;IF ZERO GO TEST ADDRESS
13107         LD     DE,FM          ;UR CALL MEMORY LOCATION
13108         INC    HL             ;SKIP SSID FOR NOW
13109         ADD    HL,BC          ;NEXT RPTR CALL LOCATION
13110         JP     CAL            ;GO TEST NEXT RPTR CALL
13111 CALRPT  LD     HL,(RCTL)      ;END REPEATER CALLS + 1
13112         LD     DE,(BGNRPT)    ;BEGIN RPTR MEM LOCATION
13113         OR     A              ;CLEAR CARRY FLAG
13114         SBC    HL,DE          ;RTCTL MINUS BEGIN RPTR
13115         JP     Z,TESADR-1     ;IF ZERO, TEST ADDRESS
13116         EX     DE,HL          ;REPEATER BYTES TO DE
13117         LD     HL,7           ;6 CALL LETTERS + SSID
13118         CALL   2490H          ;DIVIDE HL INTO DE
13119         CALL   0A7FH          ;SINGLE PREC. TO INTEGER
13120         LD     A,L            ;NUMBER REPEATER CALLS
13121         LD     (NUMRPT),A     ;STASH THEM IN MEMORY
13122         RET                   ;GOTO LINE 12530
13123 BGNRPT  DEFW   0              ;SAVE 2 BYTES FOR ADDRESS
13124 NUMRPT  DEFB   0              ;SAVE 1 BYTE RPTR COUNT
13130 REXIT   LD     IY,37873       ;DISPLAY <FORWARDING>
13140         CALL   SHOWIT         ;MESSAGE ON VIDEO
13150         LD     DE,(ENDIT)     ;CLOSING FLAG ADDRESS
13160         LD     HL,(REX)       ;MID-MEM BEGIN ADDRESS
13170         PUSH   HL             ;SWAP HL
13180         POP    IY             ;INTO IY
13190         LD     A,(FRMNUM)     ;FRAMES/PACKET COUNTER
13200         CP     0              ;SINGLE FRAME PACKET ?
13210         CALL   Z,SFRM1        ;IF SO, SET FOR SINGLE
13220         CALL   NZ,SFRM2       ;ELSE SET MULTI-FRAME
13230         LD     A,1            ;SET THE XMIT SUBROUTINE
13240         LD     (LASONE),A     ;LAST ONE POINTER
13250         XOR    A              ;ZERO OUT TRANSMIT
13260         LD     (ZEROMK),A     ;MARK COUNTER
13270         LD     (ZEROSP),A     ;AND SPACE COUNTER
13280         LD     (FRMNUM),A     ;AND FRAME COUNTER TOO
13290         CALL   SETIT          ;SETUP FOR SEND7 XMIT
13300         JP     FLGDLY         ;FINALLY - GO SEND IT
```

Appendix

The following document is reprinted by permission of the Director, CCIR. Recommendation 476-3 specifies the protocol for AMTOR (Amateur Teleprinting Over Radio) and is referenced in FCC rules section 97.69.

RECOMMENDATION 476-3 *

DIRECT-PRINTING TELEGRAPH EQUIPMENT IN THE MARITIME MOBILE SERVICE

(Question 5/8)

(1970-1974-1978-1982)

The CCIR,

CONSIDERING

a) that there is a requirement to interconnect mobile stations, or mobile stations and coast stations, equipped with start-stop apparatus employing the International Telegraph Alphabet No. 2, by means of radiotelegraph circuits;

b) that direct-printing telegraphy communications in the maritime mobile service can be listed in the following categories:

b.a telegraph service between a ship and a coast station;

b.b telegraph service between a ship and an extended station (ship's owner) via a coast station;

b.c telex service between a ship and a subscriber of the (international) telex network;

b.d broadcast telegraph service from a coast station to one or more ships;

b.e telegraph service between two ships or between one ship and a number of other ships;

* The Director, CCIR is requested to bring this Recommendation to the attention of the CCI'TT

c) that those categories are different in nature and that consequently different degrees of transmission quality may be required;

d) that the categories given in *b.a*, *b.b* and *b.c* above may require a higher transmission quality than categories *b.d* and *b.e* for the reason that data could be handled through the services in the categories *b.a*, *6.6* and *b.c*, while the messages passed through the service of category *b.d*, and via the broadcast service of category *b.e* are normally plain language, allowing a lower transmission quality than that required for coded information;

e) that the service in category *b.d* and the broadcast service in category *b.e* cannot take advantage of an ARQ method, as there is in principle no return path;

f) that for these categories of service which by their nature do not allow the use of ARQ, another mode, i.e. the forward error-correcting (FEC) mode should be used;

g) that the period for synchronization and phasing should be as short as possible and should not exceed 5 seconds;

h) that most of the ship stations do not readily permit simultaneous use of the radio transmitter and radio receiver;

j) that the equipment on board ships should be neither unduly complex nor expensive;

k) that provision is made in Appendix 38 of the Radio Regulations for direct-printing telegraph operation,

UNANIMOUSLY RECOMMENDS

1. that when an error-detecting and correcting system is used for direct-printing telegraphy in the maritime mobile service, a 7-unit ARQ system or a 7-unit forward acting, error-correcting and indicating time-diversity system, using the same code, should be employed;

2. that equipment designed in accordance with § 1 should meet the characteristics laid down in Annex I.

ANNEX I

1. General (Mode A, ARQ and Mode B, FEC)

1.1 The system is a single-channel synchronous system using the 7-unit error-detecting code as listed in § 2 of this Annex.

1.2 The modulation rate on the rad10 link is 100 bauds. The equipment clocks controlling the modulation rate should have an accuracy of better than 30 parts in 10^6.

Note. – Some existing equipments may not conform to this requirement.

1.3 The terminal input must be able to accept the 5-unit start-stop CCITT International Telegraph Alphabet No. 2 at a modulation rate of 50 bauds.

1.4 The frequency shift on the radio link is 170 Hz. When frequency shift is effected by applying audio signals to the input of a transmitter, the centre frequency of the audio spectrum offered to the transmitter should be 1700 Hz.

Note. – A number of equipments are presently in service, using a centre frequency of 1500 Hz. These may require special measures to achieve compatibility.

1.5 The radio frequency tolerance of the transmitter and the receiver should be in accordance with Appendix 38 of the Radio Regulations. It is desirable that the receiver employs the minimum practicable bandwidth (see also Report 585).

Note. – The receiver bandwidth should preferably be between 270 and 340 Hz.

2. Table of conversion

2.1 *Traffic information signals*

TABLE I

Combination No.	Letter-case	Figure-case	International Telegraph Alphabet No. 2 Code	Emitted 7-unit signal (1)
1	A	—	ZZAAA	BBBYYYB
2	B	?	ZAAZZ	YBYYBBB
3	C	:	AZZZA	BYBBBYY
4	D	⊠(3)	ZAAZA	BBYYBYB
5	E	3	ZAAAA	YBBYBYB
6	F	0 2	ZAZZA	BBYBBYY
7	G	02	AZAZZ	BYBYBBY
8	H	08	AAZAZ	BYYBYBB
9	I		AZZAA	BYBBYYB
10	J	Audible signal	ZZAZA	BBBYBYY
11	K	(	ZZZZA	YBBBBYY
12	L	)	AZAAZ	BYBYYBB
13	M	.	AAZZZ	BYYBBBY
14	N		AAZZA	BYYBBYB
15	0	4	AAAZZ	BYYYBBB
16	P	0	AZZAZ	BYBBYBY
17	Q	1	ZZZAZ	YBBBYBY
18	R	4	AZAZA	BYBYBYB
19	S	,	ZAZAA	BBYBYYB
20	T	5	AAAAZ	YYBYBBB
21	U	7	ZZZAA	YBBBYYB
22	V	=	AZZZZ	YYBBBBY
23	W	2	ZZAAZ	BBBYYBY
24	X	/	ZAZZZ	YBYBBBY
25	Y	6	ZAZAZ	BBYBYBY
26	Z	+	ZAAAZ	BBYYYBB
27	← (Carriage return)		AAAZA	YYYBBBB
28	≡ (Line feed)		AZAAA	YYBBYBB
29	↓ (Letter shift)		z z z z z	YBYBBYB
30	↑ (Figure shift)		ZZAZZ	YBBYBBY
31	Space		AAZAA	YYBBBYB
32	Unperforated tape		AAAAA	YBYBYBB

(1) B represents the higher emitted frequency and Y the lower.

(2) At present unassigned (see CCITT Rec. El C8). Reception of these signals, however, should not initiate a request for repetition.

(3) The pictorial representation shown is a schematic of ✠ which may also be used when equipment allows (CCITT Rec. F.1).

2.2 *Service information signals*

TABLE II

Mode A (ARQ)	Emit ted signal	Mode B (FEC)
Control signal 1 (CS1)	BYBYYBB	
Control signal 2 (CS2)	YBYBYBB	
Control signal 3 (CS3)	BYYBBYB	
Idle signal β	BBYYBBY	Phasing signal 1
Idle signal *a*	BBBBYYY	Phasing signal 2
Signal repetition	YBBYYBB	

3. Characteristics

3.1 *Mode* A *(ARQ)* (see Figs. 1 and 2)

A synchronous system, transmitting blocks of three characters from an information sending station (ISS) towards an information receiving station (IRS), which stations can, controlled by the control signal 3 (see § 2.2). interchange their functions.

3.1.1 *Master and slave arrangements*

3.1.1.1 The station that initiates the establishment of the circuit (the calling station) becomes the "master" station, and the station that has been called will be the "slave" station;

this situation remains unchanged during the entire time in which the established circuit is maintained, regardless of which station, at any given time, is the Information Sending Station (ISS) or Information Receiving Station (IRS);

3.1.1.2 the clock in the master station controls the entire circuit (see circuit timing diagram, Fig. 1);

3.1.1.3 the basic timing cycle is 450 ms, and for each station consists of a transmission period followed by a transmission pause during which reception is effected;

3.1.1.4 the master station transmitting time distributor is controlled by the clock in the master station;

3.1.1.5 the slave station receiving time distributor is controlled by the received signal;

3.1.1.6 the slave station transmitting time distributor is phase-locked to the slave station receiving time distributor; i.e. the time interval between the end of the received signal and the start of the transmitted signal (t_E in Fig. 1) is constant;

3.1.1.7 the master station receiving time distributor is controlled by the received signal.

3.1.2 *The Information Sending Station (ISS)*

3.1.2.1 Groups the information to be transmitted into blocks of three characters (3 x 7 signal elements), including, if necessary, "idle signals β" to complete or to fill blocks when no traffic information is available;

3.1.2.2 emits a "block" in 210 ms after which a transmission pause of 240 ms becomes effective, retaining the emitted block in memory until the appropriate control signal confirming correct reception by the Information Receiving Station (IRS) has been received;

3.1.2.3 numbers successive blocks alternately "Block 1" and "Block 2" by means of a local numbering device. The first block should be numbered "Block 1" or "Block 2" dependent on whether the received control signal (see § 3.1.4.5) is a control signal 1 or a control signal 2. The numbering of successive blocks is interrupted at the reception of:

- a request for repetition; or
- a mutilated signal; or
- a control signal 3 (see § 2.2);

3.1.2.4 emits the information of Block 1 on receipt of control signal 1 (see § 2.2);

3.1.2.5 emits the information of Block 2 on receipt of control signal 2 (see § 2.2);

3.1.2.6 emits a block of three "signal repetitions" on receipt of a mutilated signal (see § 2.2).

3.1.3 *The Information Receiving Station (IRS)*

3.1.3.1 Numbers the received blocks of three characters alternately "Block 1" and "Block 2" by a local numbering device, the numbering being interrupted at the reception of:

- a block in which one or more characters are mutilated; or
- a block containing at least one "signal repetition"; (3.1.2.6)

3.1.3.2 after the reception of each block, emits one of the control signals of 70 ms duration after which a transmission pause of 380 ms becomes effective;

3.1.3.3 emits the control signal 1 at the reception of:

- an unmutilated "Block 2", or
- a mutilated "Block 1", or
- Block 1" containing at least one "signal repetition";

3.1.3.4 emits the control signal 2 at reception of:

- an unmutilated "Block 1", or
- a mutilated "Block 2", or
- a "Block 2" containing at least one "signal repetition".

3.1.4 *Phasing*

3.1.4.1 When no circuit is established, both stations are in the "stand-by" position. In this stand-by position no ISS or IRS and no master or slave position is assigned to either of the stations;

3.1.4.2 the station desiring to establish the circuit emits the "call" signal. This "call" signal is formed by two blocks of three signals;

3.1.4.3 the call signal contains:

- in the first block: "signal repetition" in the second character place and any combination of information signals * in the first and third character place,
- in the second block: "signal repetition" in the third character place preceded by any combination of the 32 information signals * in the first and second character place;

3.1.4.4 on receipt of the appropriate call signal the called station changes from stand-by to the IRS position and emits the control signal 1 or the control signal 2;

3.1.4.5 on receipt of two consecutive identical control signals, the calling station changes into ISS and operates in accordance with § 3.1.2.4 and 3.1.2.5.

3.1.5 *Rephasing* **

3.1.5.1 When reception of information blocks or of control signals is continuously mutilated, the system reverts to the "stand-by" position after a predetermined time (a preferable predetermined time would be the duration of *32* cycles of 450 ms), to be decided by the user, of continuous repetition; the station that is master station at the time of interruption immediately initiates rephasing along the same lines as laid down in § 3.1.4;

3.1.5.2 if, at the time of interruption, the slave station was in the IRS position, the control signal to be returned after phasing should be the same as that last sent before the interruption to avoid the loss of an information block upon resumption of the communication. (Some existing equipments may not conform to this requirement);

3.1.5.3 however, if, at the time of interruption, the slave station was in the ISS position, it emits, after having received the appropriate call blocks, either:

- the control signal 3; or
- the control signal 1 or 2 in conformity with § 3.1.4.4, after which control signal 3 is emitted to initiate changeover to the ISS position;

3.1.5.4 if rephasing has not been accomplished within the time-out interval of § 3.1.9.1, the system reverts to the stand-by position and no further rephasing attempts are made.

3.1.6 *Change-over*

3.1.6.1 *The Information Sending Station (ISS)*

- Emits, to initiate a change in the direction of the traffic flow, the information signal sequence "Figure shift" – "Plus" ("figure case of *Z*") – "Question Mark" ("figure case of *B*") *** followed, if necessary, by one or more "Idle Signals β" to complete a block;
- emits, on receipt of a control signal 3, a block containing the signals "Idle Signal β" – "Idle Signal α" – "Idle Signal β";
- changes subsequently to IRS after the reception of a "signal repetition".

3.1.6.2 *The Information Receiving Station (IRS)*

- Emits the control signal 3:
  - (a) when the station wishes to change over to ISS,
  - (b) on receipt of a block in which the signal information sequence "Figure shift" – "Plus" – (figure-case of Z) – "Question Mark" (figure-case of B) terminates *** or upon receipt of the following block. In the latter case, the IRS shall ignore whether or not one or more characters in the last block are mutilated:
- changes subsequently to ISS after reception of a block containing the signal sequence "Idle signal β" – "Idle signal α" "Idle signal β";
- emits one "signal repetition" as a master station, or a block of three "signal repetitions" as a slave station, after being changed into ISS;

3.1.7 *Output to line*

3.1.7.1 the signal offered to the line output terminal is a 5-unit start-stop signal at a modulation rate of 50 bauds.

3.1.8 *Answerback*

3.1.8.1 The WRU (Who are you?) sequence, which consists of combination Nos. 30 and 4 in the International Telegraph Alphabet No. 2, is used to request terminal identification.

---

* The composition of these signals and their assignment to individual ships require international agreement (see Recommendation 491).

** Some coast stations do not provide rephasing (see also Recommendation 492).

*** In the Telex network, the signal sequence combination No. 26 - combination No. 2, sent whilst the teleprinters are in the figure case condition, is used to initiate a reversal of the flow of information. The IRS is, therefore, required to keep track of whether the traffic information flow is in the letter-case or figure-case mode to ensure proper end-to-end operation of the system.

3.1.8.2 The Information Receiver Station (IRS), on receipt of a block containing the WRU sequence, which will actuate the the teleprinter answerback code generator:

- changes the direction of traffic flow in accordance with § 3.1.6.2;
- transmits the signal information characters derived from the teleprinter answerback code generator;
- after transmission of 2 blocks of "Idle signals β" (after completion of the answerback code, or in the absence of an answerback code), changes the direction of traffic flow in accordance with § 3.1.6.1.

Note. – Some existing equipments may not conform to this requirement.

3.1.9 *End of communication*

3.1.9.1 When reception of information blocks or of control signals is continuously mutilated, the system reverts to the "stand-by" position after a predetermined time of continuous repetition, which causes the termination of the established circuit, (a preferable predetermined time would be the duration of 64 cycles of 450 ms);

3.1.9.2 the station that wishes to terminate the established circuit transmits an "end of communication signal";

3.1.9.3 the "end of communication signal" consists of a block containing three "Idle Signal *a":*

3.1.9.4 the "end of communication signal" is transmitted by the ISS;

3.1.9.5 if an IRS wishes to terminate the established circuit it has to change over to ISS in accordance with § 3.1.6.2;

3.1.9.6 the IRS that receives an "end of communication signal" emits the appropriate control signal and reverts to the "stand-by" position;

3.1.9.7 on receipt of a control signal that confirms the unmutilated reception of the "end of communication signal", the ISS reverts to the "stand-by" position;

3.1.9.8 when after a predetermined number of transmissions * of the "end of communication signal" no control signal has been received confirming the unmutilated reception of the "end of communication signal", the ISS reverts to the stand-by position and the IRS times out in accordance with § 3.1.9.1.

3.2 *Mode B, forward error correction (FEC)* (see Figs. 3 and 4)

A synchronous system, transmitting an uninterrupted stream of characters from a station sending in the collective B-mode (CBSS) to a number of stations receiving in the collective B-mode (CBRS), or from a station sending in the selective B-mode (SBSS) to one selected station receiving in the selective B-mode (SBRS).

3.2.1 *The station sending in the collective or in the selective B-mode (CBSS or SBSS)*

3.2.1.1 Emits each character twice: the first transmission (DX) of a specific character is followed by the transmission of four other characters, after which the retransmission (RX) of the first character takes place, allowing for time-diversity reception at 280 ms time space;

3.2.1.2 emits as a preamble to messages or to the call sign, **alternately** the phasing signal 1 (see § 2.2) and the phasing signal 2 (see § 2.2) whereby phasing signal 1 is transmitted in the RX, and phasing signal 2 in the DX position. At least four of these signal pairs (phasing signal 1 and phasing signal 2) should be transmitted.

3.2.2 The station sending in the collective B-mode (CBSS)

3.2.2.1 Emits during the breaks between two messages in the same transmission the phasing signals 1 and the phasing signals 2 in the RX and the DX position, respectively.

3.2.3 The station sending in the selective B-mode (SBSS)

3.2.3.1 Emits after the transmission of the required number of phasing signals (see § 3.2.1.2) the call sign of the station to be selected. This call sign is a sequence of four characters that represents the number code of the called station. This transmission takes place in the time diversity mode according to § 3.2.1.1;

3.2.3.2 emits the call sign and all further signals in a 3B/4Y ratio, i.e. inverted with respect to the signals in Table I of § 2 in the column "emitted 7-unit signal". Consequently, **all** signals, i.e. both traffic information signals and service information signals, following the phasing signals are transmitted in the 3B/4Y ratio;

3.2.3.3 emits the service information signal "Idle signal β" during the idle time between the messages consisting of traffic information signals.

* A preferable predetermined number would be four transmissions of the "end of communication signal".

3.2.4 The station(s) receiving in the collective or in the selective B-mode (CBRS or SBRS)

3.2.4.1 Checks both characters (DX and RX), printing an unmutilated DX or RX character, or printing an error symbol or space, if both are mutilated.

3.2.5 Phasing

3.2.5.1 When no reception takes place, the system is in the "stand-by" position as laid down in § 3.1.4.1;

3.2.5.2 on receipt of the sequence *"phasing signal 1" – "phasing signal 2"*, or of the sequence *"phasing signal 2" – "phasing signal 1"*, in which phasing signal 2 determines the DX and phasing signal 1 determines the RX position, and at least one further phasing signal in the appropriate position, the system changes from "stand-by" to the CBRS position;

3.2.5.3 when started as CBRS the system changes to the SBRS (selectively called receiving station) position on receipt of the inverted characters representing its selective call number;

3.2.5.4 having been changed into the CBRS or into the SBRS position the system offers continuous stop-polarity to the line output terminal until either the signal "carriage return" or "line feed" is received;

3.2.5.5 when started as SBRS, the decoder re-inverts all the following signals received to the 3Y/4B ratio, so that these signals are offered to the SBRS in the correct ratio, but they remain inverted for all other stations;

3.2.5.6 both the CBRS and the SBRS revert to the stand-by position if, during a predetermined time, the percentage of mutilated signals received has reached a predetermined value.

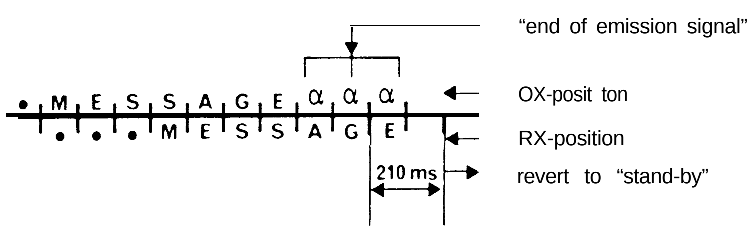
3.2.6 Output to line

3.2.6.1 The signal offered to the line output terminal is a 5-unit start-stop CCITT International Telegraph Alphabet No. 2 signal at a modulation rate of 50 bauds.

3.2.7 End of emission

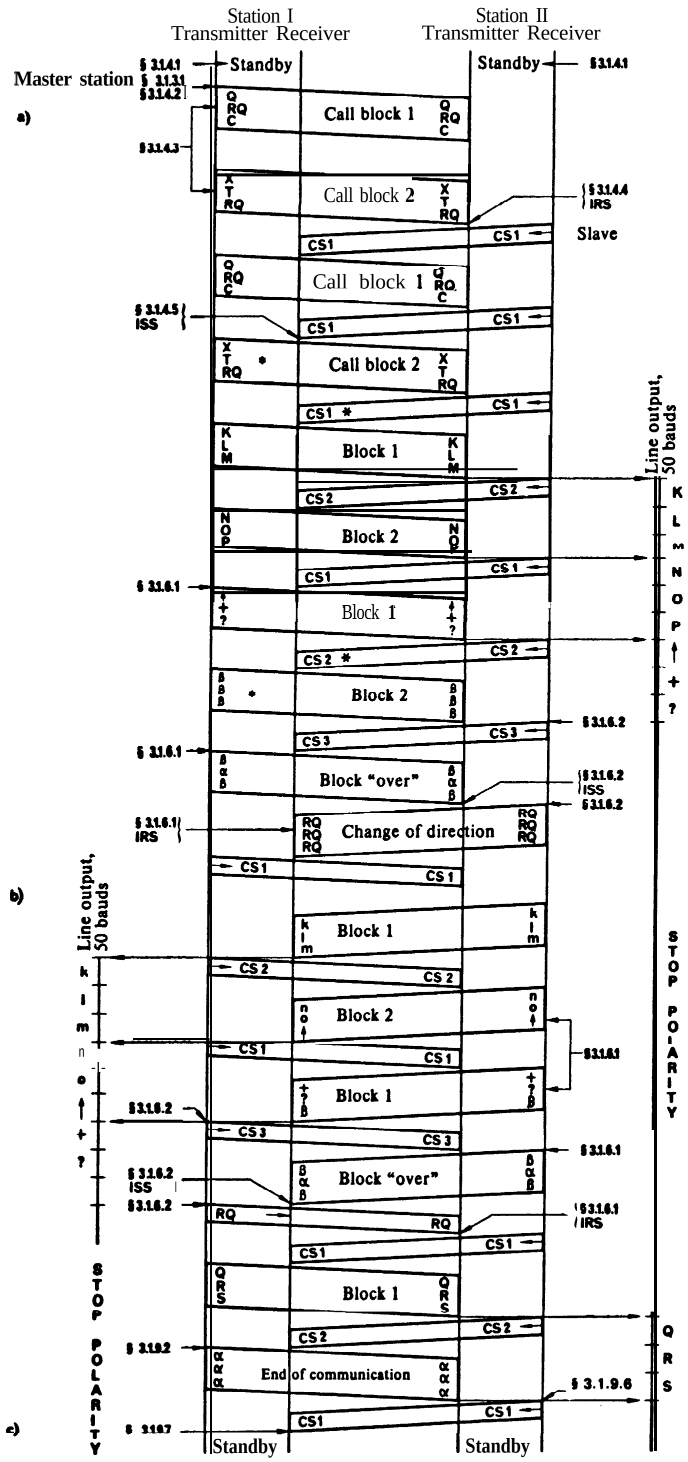
3.2.7.1 The station sending in the B-mode (CBSS or SBSS) that wishes to terminate the emission transmits the "end of emission signal";

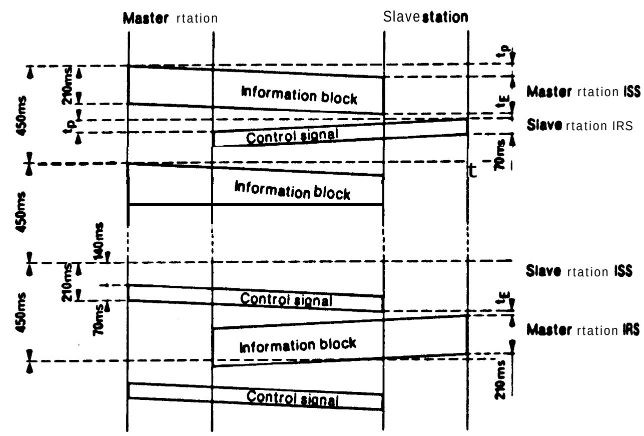
3.2.7.2 the "end of emission signal" consists of three consecutive "idle signals *α*" (see § 2.2) transmitted in the DX position only, immediately after the last transmitted traffic information signal in the DX position, after which the station terminates its emission and reverts to the "stand-by" position;



3.2.7.3 the CBRS or the SBRS reverts to the "stand-by" position not less than 210 ms after receipt of at least two consecutive "idle signals α" in the DX position.

Selective call No. 32610 transmitted as
(see Rec. 491 § 2 ,3) Q(RQ)C XT(RQ)





Basic timing cycle

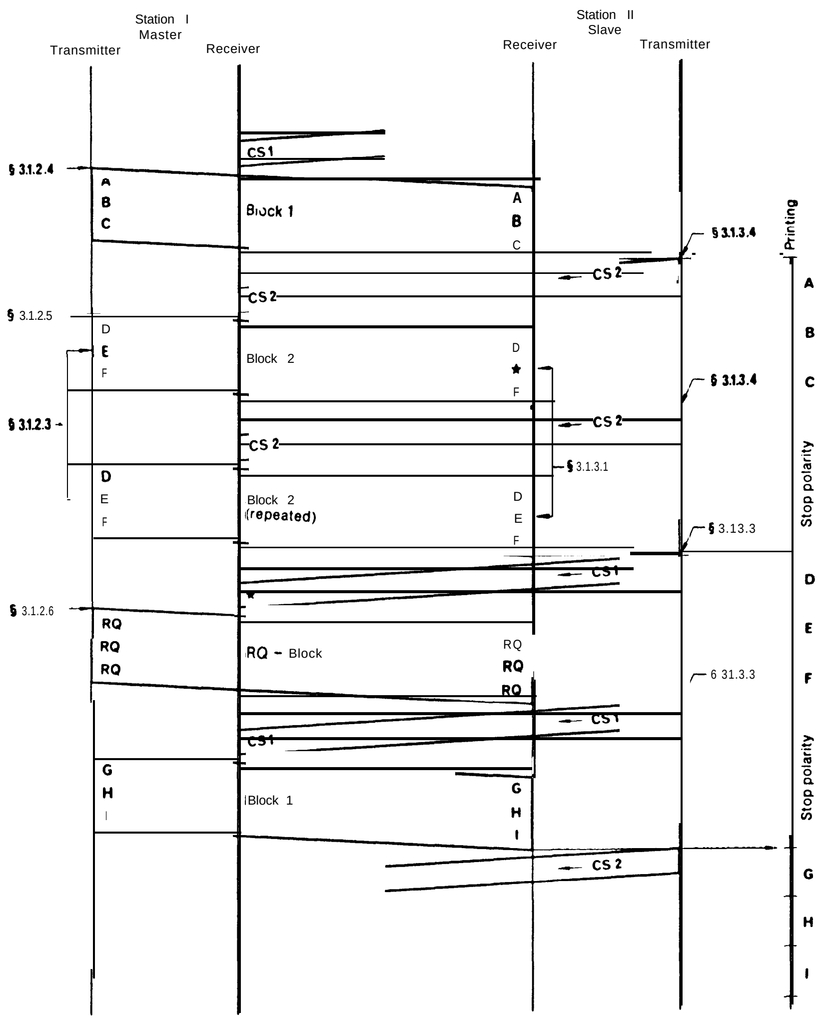
FIGURE 1 – *A-Mode operation*

a) Start of communication
b) Change of the direction of the traffic flow
c) End of communication
CS: Control signal

ISS : Information sending station
IRS : Information receiving station
RQ: Signal repetition information signal

t: Figure shift
t_p: (One way) propagation time
t_E: (Fixed) equipment delay

*The transmission of these signals may be omitted.



• Detected error Symbol.

FIGURE 2 – *Mode A under error receiving conditions*

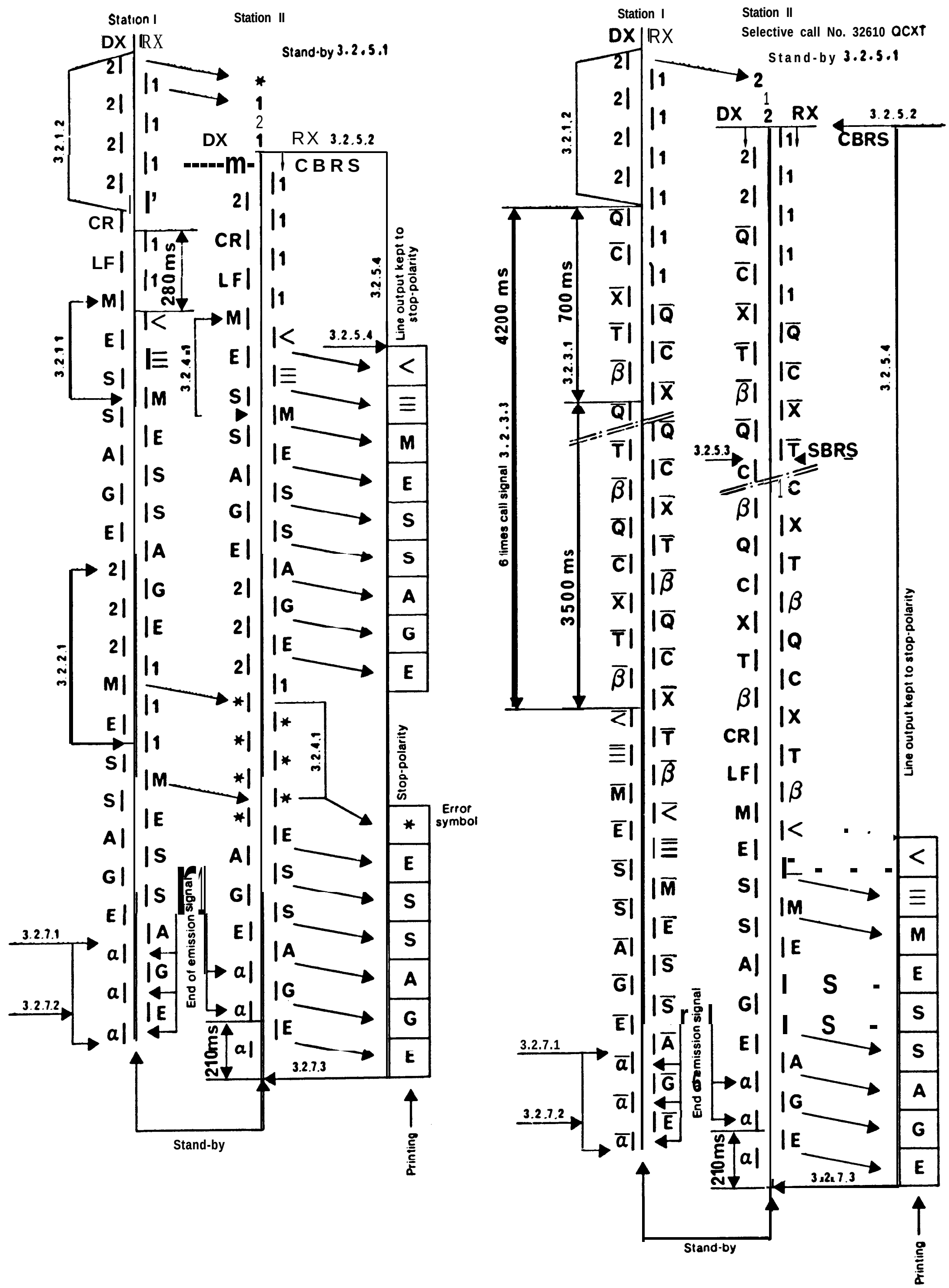


FIGURE 3 – *B-mode operation*

Collectively

1 : Phasing signal 1
2: Phasing signal 2
<: Carriage return (CR)
▬: Line feed (LF)
● : Detected error symbol

Selectively

CBSS: B-mode – Sending collectively
CBRS: B-mode – Receiving collectively
SBSS: B-mode – Sending selectively
SBRS: B-mode – Receiving selectively

Overlined symbols (e.g. $\overline{M}$) are transmitted in the 3B/4Y ratio

Fourth ARRL Amateur Radio Computer Networking Conference

March 30, 1985
San Francisco

Coordinators:
Paul L. Rinaldo, W4RI
Dr. Henry S. Magnuski, KA6M

COMMUNICATIONS PROTOCOLS FOR THE NETWORK AND TRANSPORT LAYERS OF THE AMATEUR PACKET NETWORK

3. Gordon Beattie, Jr., N2DSY
Thomas A. Moulton, W2VY

The Radio Amateur Telecommunications Society
206 North Vivyen Street
Bergenfield, NJ 07621
201-387-8896

ABSTRACT

There has been much discussion among amateurs about internetworking with other areas of the country and globe. This has led to the introduction of terms into the vocabulary of many amateurs, many of whom are newly equipped with computers ! In this paper we will present the ISO/CCITT Open Systems concept and its impact on the protocols we will be using to provide reliable data transfer in the amateur network. In order to provide a basis for contrast, we will also introduce the U.S. Department of Defense protocols.

First we will define the basic concepts and terms needed to understand the protocol issues. Once this groundwork is laid, we will explore the protocols needed to provide a reliable data transfer capability. Conclusions will be drawn based upon the experience of the writers and their amateur and professional associates.

TERMS

OS1 -
Open Systems Interconnection Model
The goal of "open systems" is to allow dissimilar computers, networks and terminals to operate together in a common network environment. There are seven layers or functions defined in the model. They provide the basis for interoperability. Standards bodies all over the world have evolved in an unprecedented way to Jointly develop these standards for the telecommunications and computer industries.

Layer -
A functional segment of the OSI model. There are seven.

Physical Layer - level 1
This layer addresses all physical aspects of the communications medium. Voltage, frequency, interface lead assignment and connector layout are all determined by physical layer standards.

Link Layer - level 2
This layer defines the functions of two directly connected points in a network. Link establishment, data transfer, fiow control, error control and term ination are defined.

Network Layer - level 3
This layer defines the means to establish, maintain and terminate network connections through a series of links which compose one or more networks. Link multiplexing, flow control and sequencing of data are also defined in this layer.

Transport Layer - level 4
This layer provides the means to ensure reliable data transfer between end points in the network. Depending on the characteristics of tlhe network layer, different error handling capabilities will be required of the transport layer. The transport layer must be prepared to supply the required reliability if it is unable to obtain a reasonable grade of service from the network layer.

Session Layer - level 5
The session layer is responsible for calling upon the resources required to complete a distributed communications task via a network or networks. A simple session would be a file transfer between two stations in real-time. An example of a more complex session would be a station obtaining needed information from a group of databases in order to locate and properly conduct a series of message transfers with a dispersed group of destinations.

Presentation Layer - level 6
The presentation layer is responsible for code and format conversion between end users. Baudot to ASCII conversion would be the responsibility of this layer.

Application Layer - level 7
This layer defines message formats and procedures used by such applications as electronic mail, bullk file transfer and

remote data processing.

Virtual Circuit - (VC)
A logically-linked path through a network or networks over which data is passed between end users. They usually are setup using call establishment procedures, used for the period of data transfer, then disolved.

Datagramme -
A data unit which contains all neccessary information for routing, error control, sequencing and throughput requirements in addition to the actual user data.

Fast Select -
An X.25 Call Request with 128 bytes of user data.

CCITT -
International Consultive Committee for Telephone and Telegraph. This organization coordinates standards-making activities in the telecommunications field.

IS0 -
The International Organization for Standardization
This body along with the CCITT, formalized the Open System concept and the protocols required to perform the functions of each layer.

X.25 -
A CCITT standard for packet switched network interfaces. It includes recommendations for the first three layers of the IS0 model. AX.25 level 2 was an adaptation of the Link layer of X.25. Level 3 of X.25 provides for a "virtual circuit" to be used as a path for data transfer.

X.224 -
A CCITT standard defining the transport protocol for use in OSI applications.

TPDU - Transport Protocol Data Unit

Internet Protocol -
A datagramme-oriented networking protocol. This level 1 protocol was adopted by the U.S. Department of Defense (DoD) for use on its Advanced Research Project Agency Network (ARPANET),

TCP -
A transport protocol developed by the DoD for ARPANET.
This level 4 protocol was developed to provide end-to-end error recovery in a datagramme-based network.

THE REAL ISSUES

The OS1 model defines peer-level protocols to be used in corresponding communicating entities. The OSI concept separates the issues before us into two ciistinct tasks:

1. The selection of a network layer (level 3) protocol.
   The network layer protocol options are the ISO/CCITT X.25 level 3 and the U.S. DoD Internet Protocol.

2. The selection of a transport layer (level 4) protocol.
   The transport layer options are the ISO/CCITT X.224 and the U.S. DoD Transmission Control Protocol (TCP).

Diagramme 1 shows where these protocols fit in the OSI scheme of things.

THE NETWORK LAYER

The first area of *concern is* the dissimilar way these protocols pass data through a network. The X.25 level 3 protocol has three phases: call set-up, data transfer and call release. A station sets up a path over which data is passed, then transfers the data. The protocol exchange for a virtual circuit (VC) based data transfer would be as shown below.

The DoD Internet Protocol uses datagrammes to provide the needed control to route the packets through the network. There is *no* '@connection" between network the sender and receiver of the the network layer data. Upper layers provide this, leaving the network layer with the tasks of routing. As we will see, this puts a large burden on the end-users of the network and limits throughput by increasing overhead.

The basic issues in deciding between the ISO/CCITT X.25 level 3 and U.S. DoD Internet Protocol are:

1. Protocol Efficiency;
2. Reliability;
3. Ease of Implementation;
4. Interoperability with Non-Amateur Networks:
5. International Acceptance.

1. Protocol Efficiency

The set-up or call establishment procedure allows all stations involved in the call to negotiate the parameters needed to transfer the data over the VC. This is accomplished through the Call Request and Call Accept exchange. It also identifies the end points to all stations involved so that each packet exchanged with user data need only contain a logical identifier. Each station supporting the call must maintain a table entry reflecting the path through the station itself and the current state of that call.

During call establishment, parameters are exchanged which govern the data flow of that call. Some of these parameters are outlined below:

Packet Window Size -
Lets each station know how many unacknowledged packets may be outstanding before an acknowledgement is required.

Packet Size -
Determines the amount of data in each packet. It must be set to a power of two <e.g. 128/256/512/1024 etc.),

Throughput Class -
An indication of the required level service needed to support the call.

Extended Address -
A way to indicate the actual stations *in* the source and destination networks using the locally defined address (e.g. W2VY-5).

The exchange in diagramme 2 shows end-to-end significance to the packet sequence numbers, but at the option of the network it may be locally acknowledged. The Call Request and Accept packets may also contain up to 128 bytes of user data.

Datagrammes contain source and destination addresses, service type, and time-to-live information in each packet. This information is used by the packet switches to route, and if necessary discard, the packet. Most packet switches in commercial networks require 8-15 times the CPU resources to handle routing functions than they do for simple data transfer across an estblished path.

The datagramme approach allows for data units to be sent over random paths causing them to often arrive out of sequence. This is done to increase reliability in the network, but such a measure is unneccessary and forces the transport layer to handle sequence management, complicating the functions of the end user terminals.

The ammount of overhead in the Internet Header is extensive when compared to that of the X.25 level 3 header. There are two cases which require examination:

1. Datagramme vs. Fast Select Packet
2. Datagramme vs. Data Packet.

An X.25 Fast Select call is a normal X.25 call with up to 128 bytes of user data included. It may be acknowledged with either an accept or a clear with up to 128 bytes of user data. In either case it mirrors the function of the datagramme protocol by creating independent data units while at the same time allowing for VC-based exchanges using the same protocol sequence and network.

An IP datagramme will usually contain 20 bytes of overhead in each packet.

The X.25 Fast Select Call Packet will typically contain twenty-four (24) bytes of overhead.

The X.25 Data Packet will contain three (3) bytes of overhead.

The format of a typical Fast Select packet would be as shown below.

```
             X.25 Call Request
             ---- ---- -------

------------------------------------
|     GFI        |      LCGN       |
------------------------------------
|              LCN                 |
------------------------------------
| Packet Type - Call Request       |
------------------------------------
| Calling Length | Called Length   |
------------------------------------
|         Called Address           |
\                                  \
|                                  |
------------------------------------
|         Calling Address          |
\                                  \
|                                  |
------------------------------------
|        Facilities Length         |
------------------------------------
|           Facilities             |
\                                  \
|                                  |
------------------------------------
|           User Data              |
\                                  \
|        up to 128 bytes           |
------------------------------------
```

Data packets exchanged after the call is established use the format found in diagramme 4.

The format of a typical datagramme packet would be as shown below.

Internet Protocol Datagramme

```
-------------------------------------
|   Version        |     IHL        |
-------------------------------------
|         Type of Service           |
-------------------------------------
|         Total Length              |
-------------------------------------
|         Total Length  (cont.)     |
-------------------------------------
|              ID                   |
-------------------------------------
|              ID (cont.)           |
-------------------------------------
|   Flag         | Fragment  Offset |
-------------------------------------
| Fragment Offset (cont.)           |
-------------------------------------
|             Time                  |
-------------------------------------
|             Protocol              |
-------------------------------------
|    Header  Checksum               |
-------------------------------------
|    Header Checksum (cont.)        |
-------------------------------------
|       Source  Address             |
-------------------------------------
|       Source  Address  (cont.)    |
-------------------------------------
|       Source  Address  (cont.)    |
-------------------------------------
|       Source  Address  (cont.)    |
-------------------------------------
|       Destination  Address        |
-------------------------------------
|       Destination Address (cont.) |
-------------------------------------
|       Destination  Address (cont.)|
-------------------------------------
|       Destination  Address (cont.)|
-------------------------------------
|       Options                     |
\                                   \
|                                   |
-------------------------------------
|   Data                            |
\                                   \
|                                   |
-------------------------------------
```

2. Reliability

Packet lengths as noted in the previous section are an important factor in reliability. If the packet is longer, it is more likely to experience errors in transmission.

The data transfer procedures in X.25 level 3 provide for properly sequenced data packets exchanged in a flow controlled-manner between end points. The sequencing of data in the network layer also allows us to eliminate the error recovery *baggage* needed in the transport layer to resequence the packets. This is a further shortening of the overhead obtained by the VC approach.

Another frequently heard cry is "What happens if a switch goes down ?" In any network there would be a timeout followed by an error recovery procedure. In the case of a datagramme network there would be a new datagramme sent. In the case of *an* X.25 network there would be Clear from the stations adjacent to the one in trouble. This would be followed by a Fast Select Call Packet. Since a definitive action has occurred, the network can decide how subsequent calls should be handled. The clearing cause and diagnostic codes provide the network status information. This information should be used to guide subsequent calls through the network.

3. Ease of Implementation

It is often suggested that it is "too much" to expect that a station or packet switch maintain tables of active virtual circuits passing through. I think we can trust our small (and someday large) switches to maintain tables of call status.

Programming an implementation of the network layer is only a step more difficult than the link layer of AX.25 Level 2. As a point of interest, Phil Karn's (KA9Q) implementation in the C language supports multiple logical links across one or more physical ports. The programming techniques needed at the network layer are almost identical to his approach to the link layer.

4. Interoperability with Non-Amateur Networks

The ability to use the same network layer procedures through interfaces between amateur and non-amateur networks would greatly simplify and expedite completion of such facilities. These interfaces would provide the data equivalent of autopatch and reverse patch. The advantage of the data system over traditional voice telephone autopatches is the presence of the computer gear to validate authorized ham access to the facility. This will go *a* long way toward keeping the FCC happy about amateur control of a non-amateur access.

The addressing format of X.121 IS usually found in X.25. It defines formats for accessing the telephone and telex networks from packet switched networks.

The X.25 protocol has source and destination address facilities which allows for address conventions to be network or user specified. There is no

addressing format in IP. The format and actual addresses allowed under TCP are assigned in groups of binary numbers. These have little or no logical relationship to the world an amateur is accustomed.

5. International Acceptance

The X.25 network layer has been impiemented in nearly every country. It is recognized and understood by most regulatory authorities and therefore lends itself to acceptance by those authorities for use in the amateur service. The U.S. DoD (remember: Department of Defense) may not be a big seller in other countries when a recognized international protocol is available.

We in the United States have witnessed the impact we've had on packet radio worldwide. We must keep in mind the needs of amateurs abroad. Their cooperation is needed to provide transit network connections and uniform end-user protocols.

THE TRANSPORT LAYER

The transport protocol options before the amateur community are ISO/CCITT X.224 and the U.S. DoD Transmission Control Protocol (TCP). Unlike their network layer cousins, these protocols have many similarities.

During the late 1960's and 1970's the U.S. Department of Defense (DOD) required networking standards for its own Advanced Research Projects Agency Network (ARPANET). At this time the international standardization effort was just being to be envisioned and no formal acitvity had commenced. The DoD developed a set of protocols for use on ARPANET.

During the last two study periods, the CCITT and IS0 have developed the needed standards. The need for proprietary protocols for general network applications is no longer Justified or constructive for cost effective network development. X.224 provides for a range of network error recovery situations. It is divided into classes of operation. For this discussion we will address class 1.

Class 1 is for use in networks with low undetected error rate, but having high rates of signaled failures. An amateur packet network using X.25 level 3 would certainly fall into this catagory. A packet network with IP as a level 3 protocol would require X.224 class 4 or TCP to provide the needed error control.

The transport layer provides for several functions in the OSI model. They are:

1. End-to-end data integrity;
2. Network connection re-establishment:
3. Re-synchronization after network layer reset.

The same criteria used in evaluating network layer protocols will be used to help us through the transport layer. They are:

1. Protocol Efficiency;
2. Reliability:
3. Ease of Implementation;
4. Interoperability with non-Amateur Networks;
5. International Acceptance.

1. Protocol Efficiency

The TCP and x.224 protocols are functionally similar, however X.224 is leaner and is better suited to the needs of the amateur network. By "leaner" we mean that it uses fewer bytes in its control header, thereby reducing the chance of error on the radio channel.

The X.224 CR TPDU will usually contain 7 bytes of overhead.

The X.224 DT TPDU will have 3 bytes of overhead.

A TCP message will usually have 20 bytes of overhead.

Again we have a situation where the set-up in the CCITT/ISO protocol uses a longer header during the establishment phase in order to gain economies during data transfer. The X.224 class 1 connection request format is as shown below.

X.224 Connection Request

Length Indicator	
TPDU - CR	CDT
Destination Reference	
Destination Reference (cont.)	
Source Reference	
Source Reference (cont.1	
Class/Options	
User Data	

X.224 class 1 operation uses the following format for data transfer:

X.224 Data Transfer

Length Indicator
TPDU - DT
Number Received
User Data

The format of a typical TCP message would be as shown below.

TCP Message

Source Port	
Source Port (cont.1	
Destination Port	
Destination Port (cont.)	
Sequence Number	
Sequence Number (cont.)	
Sequence Number (cont.)	
Sequence Number (cont.)	
Acknowledgement Number	
Acknowledgement Number (cont.)	
Acknowledgement Number (cont.)	
Acknowledgement Number (cont.)	
Data Offset	Reserved
Reserved	Control Bits
Window	
Window (cont.)	
Checksum	
Checksum (cont.)	
Urgent Pointer	
Urgent Pointer (cont.1	
Options	
Data	

2. Reliability

Both protocols are deemed reliable for amateur service. It should be noted however, that TCP has a much larger overhead. Any additional functions in TCP that are not used in X.224 class 1 operation are just not needed given a VC-based network.

3. Ease of Implementation

Implementation of either protocol would require the same general tasks. There are fewer implementations of X.224 at this time but this will change. I know of several completed implementations and many more are under development; It is projected to be more widespread than TCP sometime during the 1986-87 time-frame.

4. Interoperability with Non-Amateur Networks

There are implementations of TCP in most Unix systems. Implementations of X.224 are increasing in popularity. Even in the U.S. DoD the CCITT/OSI protocols are attaining new-found recognition. In the February, 1985 issue of Data Communications Magazine, Jerrold S. Foley wrote:

"The U.S. Department of Defense (DOD) has been reviewing OSI protocols for suitability to its requirements. Since both the DOD TCP <transport control protocol> and the OSI transport protocol have strong ties to the Defense Advanced Research Projects Agency, the prospect of interoperability or REPLACEMENT (caps author) of TCP is good. The North Atlantic Treaty Organization has indicated that OSI is central to its data communications planning."

That is certainly interesting food for thought...

5. International Acceptance

TCP is being used world-wide on the ARPANET and in Unix mail applications, but X.224 is being used to support a great number of videotext terminals around the world. There are other classes of operation available within X.224. They can be used with little change to the end user software if the need arises. One such application would be multiplexing many transport connections onto a single network connection. In this case the use of X.224 class 3 would be in order.

SUMMARY

Overall, the protocols available from the CCITT/ISO seem superior to those in

the DoD arsenal. The following characteristics stand out:

1. Header Length
   In data transfer, the header length of a TCP/IP packet is at least 40 bytes. The header length in a X.25 level 31X.224 class 1 packet is 7 bytes.

2. Data Parsing in Packet Switches
   The TCP/IP header must be examined by each station in the path to ensure proper handling and routing. This increases switch overhead by 8-15 times above that of the vc switch.

3. U.S. DOD or CCITT/ISO
   NATO and the U.S. DoD have made statements of direction leading to the replacement of TCP with CCITT/ISO protocols on the ARPANET and the Defense Data Network (DDN).

4. Virtual Circuit vs. Datagramme
   No commercial network offers datagrammes. Datagrammes were dropped from X.25 during the 1984 study period,

5. Who has the knowledge to implement a VC based switch ?

   Existing amateur packet implementations have more than demonstrated the availability of programming talent capable of implementing a VC-based packet switch.

6. Packet Voice
   Packet voice has been successfully implemented over virtual circuit networks. The need to transmit data over a network in a short period of time is well handled by CCITT/ISO protocols. The British Telecom network and Japan's KDD have throughput class implemented as a standard offering and GTE Telenet will be Providing voice packet service in late 1985.

7. Interconnection with Public Data and Telephone Networks
   National and international interconnections to public telephone, data and telex services will greatly depend upon the use of internationally recognized protocols.

8. Excess Protocol Baggage
   TCP offers error recovery from problems which we should not have if we use a VC-based network protocol. The same error recovery is available from X.224, but is not a required function of all implementations.

CONCLUSION

Let us understand that the marketplace has already made a stand which is driving the packet switching industry, user community and the standards bodies. Commercial and private packet switched networks are overwhelmingly using Virtual Circuit (VC) based protocols, not datagrammes. This is not to say that the amateur community is bound to follow commercial standards, but it certainly tilts the scales a bit, The Radio Amateur Telecommunications Society wishes to continue in the direction of CCITT/ISO protocols. This direction was started over three years ago in Joint meetings with the Amateur Radio Developement Corporation. These meetings culminated in the ARRL's adoption of AX.25 Level 2 as the amateur link protocol. This standard has done mugch to stimulate and direct packet radio activities to the more complex issues described in this paper and others in the procedings. We hope that the combined efforts of the amateur community will continue to yield great results in the future.

ACKNOWLEDGEMENTS

The authors' wish to thank Hal Folts, Terry Fox WB4JFI, Howie Goldstein N2WX, John Howell, Phil Karn KA9Q (for his "War" coverage and reports via DR.NET), Paul Newland ad7i, Brian Riley KA2BQE, and Nancy Beattie (for proofing the copy and turning her dinning room into a study hall).

OFFER

In keeping with its active support of CCITT/ISO protocols, the Radio Amateur Telecommunications Society (RATS) will send to interested parties copies of the protocol documents listed below. Call or write to make arrangements. RATS reserves the right to withdraw this offer at any time without notice.

X.25 - Packet Switched Network Interface Specification

X.200- Reference model of Open Systems Interconnection for CCITT applications

X.210- OSI layer service definition conventions

X.213- Network service definition for Open Systems Interconnection for CCITT applications

X.214- Transport service definition for Open Systems Interconnection for CCITT applications

X.215 Session service definition for Open Systems Interconnection for CCITT applications

X.224- Transport protocol specification Open Systems Interconnection for CCITT applications

X.225- Session protocol specification Open Systems Interconnection *for* CCITT applications

X.2440 Procedure for the exchange of protocol identification during virtual call establishment on packet switched public data networks

X.250- Formal description techniques for data communications protocols and services.

Diagramme 1
--------- -

```
                OPEN  SYSTEM  INTERCONNECTION  REFERENCE  MODEL
                ----  ------  --------------  ---------  -----

Level     END USER A                                 END USER B
          ----------------                           ----------------
  7       | Application  |  <------------------>     | Application  |
          ----------------                           ----------------
  6       I Presentation |  <------------------>     | Presentation |
          ----------------                           -----W-W--------
  5       | Session      |  <------------------>     t Session      |
          ----------------                           ----------------
  4       | * TRANSPORT *I  <------------------>     | * TRANSPORT *|
          ----------------                           ----------------
  3       | *  NETWORK * |  <------------------>     | *  NETWORK * |
          ----------------                           ----------------
  2       | Link         |  <------------------>     | Link         |
          ----------------                           ----------------
  1       I Physical     |  <------------------>     I Physical     |
          ----------------                           -----------I----
```

Diagramme 2

Protocol Exchange: *X.25 Level 3*

Source	Station A	Station B	Destination
Call -->			
	Call -->		
		Call -->	
			Call -->
			<-- Accept
		<-- Accept	
	<-- Accept		
<-- Accept			
Data 0 -->			
Data 1 -->	Data 0 -->		
	Data 1 -->	Data 0 -->	
		Data 1 -->	Data 0 -->
			<-- RR 0
		<-- RR 0	Data 1 -->
Data 2 -->	<-- RR 0		
<-- RR 0	Data 2 -->		
		Data 2 -->	
			Data 2 -->
			<-- RR 2
		<-- RR 2	
Data 3 -->	<-- RR 2		
<-- RR 2		Data 3 -->	
			Data 3 -->
			<-- RR 3
		<-- RR 3	
	<-- RR 3		
<-- RR 3			
Clear -->			
	Clear -->		
		Clear -->	
			Clear -->
			<-- Clr Cfm
		<-- Clr Cfm	
	<-- Clr Cfm		
<-- Clr Cfm			

Diagramme 3

```
                Protocol Exchange:  Internet Protocol
                -------- ---------  -------- --------

Source       Station A    Station C   Station B      Destination
-----------------------------------------------------------------
Data 0 -->
Data 1 -->      Data 0 -->
                Data 1 --------------------->
                          Data 0 -->
                                    Data 0 -->       Data 1 -->
                                                     Data 0 -->
                                                     <-- ACK 0
                                    <-- ACK 0
                          <-- ACK 0
                <-- ACK 0
<-- ACK 0
Data 2 -->
                Data 2 -->
                          Data 2 -->
                                    Data 2 -->
                                                     Data 2 -->
                                                     <-- ACK 2
                                    <-- ACK 2
                          <-- ACK 2
               <-- ACK 2
<-- ACK 2
```

Diagramme 4

```
                      X.25 Data Packet
                      ---- ---- ------

               --------------------------------------
               |      GFI        |      LCGN        |
               --------------------------------------
               |                LCN                 |
               --------------------------------------
               |        Packet Type - Data          |
               --------------------------------------
               |              User Data             |
               \                                    \
               |      up to 256 bytes (default)     |
               --------------------------------------
```

PROPOSAL: RECOMMENDATION AX.121NA

NUMBERING PLAN FOR THE AMATEUR RADIO NETWORK IN NORTH AMERICA

J. Gordon Beattie, Jr., N2DSY
Thomas A. Moulton, W2VY

The Radio Amateur Telecommunications Society
206 North Vivyen Street
Bergenfield, NJ 07621
201-387-8896

Introduction

The purpose of this Numbering Plan is to facilitate the introduction of amateur data networks and provide for internetworking in the North American region.

1.0 Design Considerations

1.1 This proposal does not require, nor preclude, governmental involvement in network administration.

1.2 The Regional Numbering Plan should permit the identification of a called country as well as a specific network and station.

1.3 The Numbering Plan should provide a consistent addressing format when connection is made with or through commercial telephone networks.(i.e. telephone, telex, data networks.)

1.4 A national number assigned to a terminal should be unique within a particular network. This national number should form part of the international number which should also be unique on a worldwide basis.

1.5 Specific national numbers should be easily determined.

1.6 National Numbers shouid require minimal administrative overhead to network management and users.

2.0 Numbering System

2.1 The 10 digit numeric character set O-9 should be used for numbers (or addresses> assigned to terminals in the amateur network. This principle should apply to both national and international numbers.

2.2 Use of the numbering system as outlined in 2.1 will make it possible to interwork with terminals on public telephone, telex and data networks.

3.0 Prefix Codes

3.1 The Prefix Code will signify the type of network indicated by the remaining digits.

3.2 The Prefix Code will be the first digit and should be coded as follows:

```
0     Amateur Packet Switched Network
1     Public Packet Switched Network
2     \
3      \
4       \___ Reserved
5       /
6      /
7     /
8     Telex Network
9     Telephone
```

4.0 Data Network Identification Codes

4.1 All Data Network Identification Codes shall consist of four digits.

4.2 Each country in the region shall use the codes listed below.

3020	Canada
3100	United States of America
3300	Puerto Rico
3320	Virgin Islands (USA)
3340	Mexico

5.0 National Number

5.1 The National Number shall consist of up to 10 digits.

5.2 Each National Number shall be unique within the country.

5.3 The National Number shall contain a three digit area code.

5.4 This number shall correspond to the area code used in the North American Numbering Plan for Telephone Networks.

5.5 Additional addressing information may be provided in an address extension facility containing the amateur callsign and SSID.

5.6 If full 10-digit addressing is desired, the number corresponding to the local exchange and subscriber line may be used.

5.7 If no number is available or if additional numbers are required they should be assigned using exchange numbers in the range of 000 through 199.

5.8 The assignment authority for these exchange and subscriber numbers is limited to the Network Coordinating Agent for that area.

5.9 Service Codes 011, 111, 211, 311, 411, 511, 611, 711, 811, 911 are reserved pending definition by the local Network Coordinating Agent.

5.10 The exchange code 000 is reserved for internal network administration and assignment authority is limited to the National Network Coordinating Agent.

5.11 The exchange code 555 is reserved for internal network administration and assignment authority is limited to the local Network Coordinating Agent.

5.12 The exchange and subscriber code 555-1212 is reserved for regional directory service. Assignment authority is limited to the local Network Coordinating Agent.

6.0 International Number

6.1 The International Number shall consist of the DNIC and the National Number.

6.2 Each National Amateur Network will be capable of interpreting the first four digits of the International Number. This is needed to facilitate routing between networks.

6.3 The use of the DNIC by stations in transit countries would serve as a ready reference for checking third-party traffic-handling requirements.

6.4 The International Number may optionally include a prefix code.

7.0 Formats

7.1 Amateur Network Number Format

P-DDDD-AAA-EEE-NNNN = 15

DDDD-AAA-EEE-NNNN = 14

7.2 Amateur Network Number Format (alternate)

P-DDDD-AAA = 8

DDDD-AAA = 7

Where:

P = Prefix digit
D = Data Network Identication Code
A = Area Code
E = Exchange
N = Number

THE FREQUENCY AGILE MESSAGE SYSTEM (FAMS)

David W. Borden, K8MMO
Amateur Radio Research and Development Corp.
Route 2, Box 233B
Sterling, VA 22170
703-450-5284

Abstract

With the increasing traffic appearing on local two meter packet radio channels, computer packet radio message systems appear to the casual conversationalist typer as channel hogs. The message systems interfere in two modes. First, a typer user connects with it and downloads large packets of file data, help messages, directory, etc. Second, a semi-automatic store and forward mode is invoked periodically to forward messages further up the network to other messages systems. The service these message systems provide more than justifies their existence, so the answer is not to ban them. A new method might alleviate the interference and retain their useful features. This method involves frequency agility, the ability to switch frequencies on command, to pass traffic.

Introduction

In the coming year, 9600 baud packet radio backbone network traffic passing on 220 MHz will remove the message computer interlinking from the main local area channel. This will alleviate interference only slightly. The casual typer will still want to read the bulletin board output and thus cause heavy contention for other typer-to-typer conversations. The basic concept is then that the casual typer, upon connecting to one of these message systems, commands the system to switch frequency to work with the typer. The message system automatically returns to the local area calling channel upon disconnect or timeout for lack of typer user activity.

The Basic System

The Xerox 820 has proved itself to be a valuable addition to the packet radio inventory. A small hardware addition (see AMRAD Newsletter, Nov/Dec 84) allows packets to be sent and received from the onboard SIO. Using software developed by Phil Karn, KA9Q, it serves as a Terminal Node Controller (TNC) and digipeater. Using software developed by Jon Bloom, KE3Z, it serves as a two port digipeater, a novel device linking two packet channels. Various amateurs have written good message system software, the most famous includes interlinking provisions for middle of the night transfers between systems. Software and hardware have been developed to change the frequency of a two meter transceiver using the Xerox 820 user PIO parallel port (see Amrad Newsletter, February 85). A marriage of some of this software would allow a frequency agile message system which would help alleviate congestion on the local area two meter packet channel.

Concept of Operation

Picture the scenario in which packet radio user David wishes to communicate with user Howard. Using standard packet techniques a connection with user Howard is requested on the local 145.01 MHz two meter local area frequency. Howard is working late and thus does not answer David's SABM frames which fall on the ground unserviced. David is not thwarted however. He connects with Tom's message system, intending to leave a message for Howard to be read later. Instead of the usual carriage return from David producing the torrent of characters from Tom's message system, the system requests David to indicate the desired frequency of operation. The frequency is entered by David and Tom's message system shifts frequency there to pass and receive traffic from David. David then commands the message system to switch frequency to 145.03, one of the coordinated alternate packet frequencies in David's area. The message system changes to the new frequency and so does David. When David is done using Tom's system, he disconnects and the system reverts to the local area frequency 145.01 to await the next user. Old NTS users will recognize this classic scheme where a user checks into the net on net frequency and moves off frequency to pass traffic, returning when finished. Congestion on the local area typer-to-typer frequency is thus reduced.

Hardware Required

To put this concept into operation requires the following hardware:

(2) Xerox 820 Computer Boards.
(2) 5 1/4 inch Floppy Disk Drives (one each board)
(2) Power supplies
(2) Keyboards
(2) CRT Monitors
(1) Packet Interface Board (State Machine)
(1) Frequency Control Interface Board (optional)
(1) ICOM IC-2AT Transceiver modified
(1) Bell Standard 202 Modem and ICOM interface:
(600 ohm transformer, resistor and transistor)

The packet interface board contains a state machine (prom and latch) for receiveing NRZI encoded frames and recovering clock as well as a divider and flip flop for transmitting NRZI encoded data. It is completly described in the referenced article. The cost of this device is about $15 and several have been built in the Washington area. A printed circuit board is planned and may be available in the future from Tom Clark, W3IWI.

The frequency control interface board is optional as all that is really required is to add a socket to the ICOM IC-2AT and a cable to the Xerox 820 acting as the TNC. The board contains a frequency display which shows the new frequency selected by the computer.

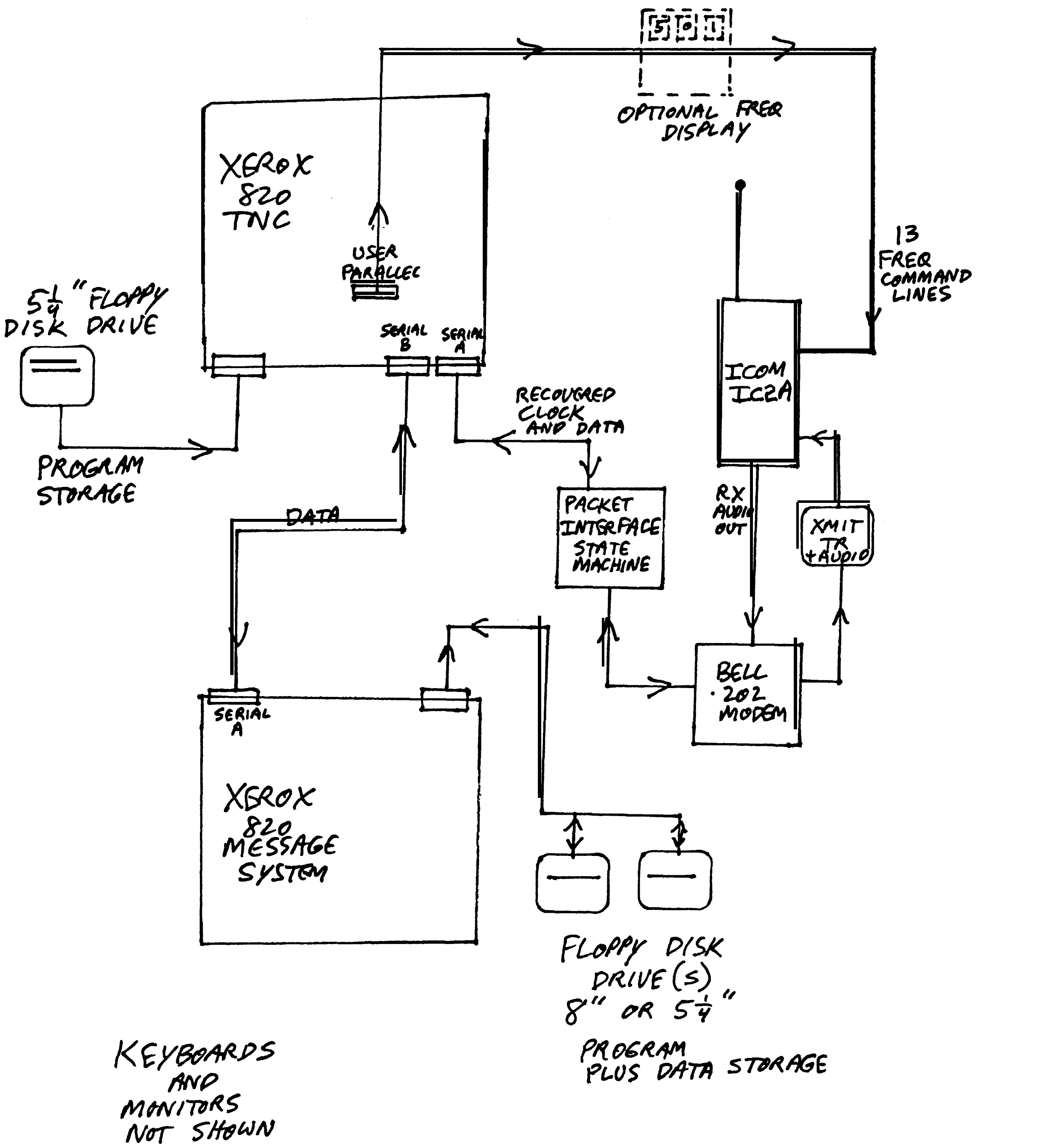
The interconnection of this hardware is as follows: The packet interface board is connected between the modem and the Xerox 820 TNC SIO port A; The frequency control interface board (or just a cable) is connected between the Xerox 820 TNC user PIO port and the ICOM IC-2AT; A serial cable connects Xerox 820 TNC SIO port B and the Xerox 820 Message System SIO port A; One power supply and one floppy disk drive is provided for each Xerox and each has a CRT and keyboard.

Software Required

The FAMS requires each Xerox run the CP/M disk operating system. AMRAD modified Phil Karn TNC code runs in the Xerox 820 TNC. Any message system software you desire runs in the Xerox 820 Message System computer. The modifications to Phil Karn's code allow command of the PIO port and setting of the ICOM frequency. In addition, Port B software of the Xerox TNC has been modified to allow terminal operations and this connects to the Xerox820 Message System computer when a packet connection is achieved on Port A.

Complete Further Details

Articles and software mentioned in this paper are obtainable from AMRAD, P.O. Drawer 6148, McLean, VA USA 22106-6148. Requests for software should be accompanied by two 5 1/4 inch floppy disks or one 8 inch floppy. Each month, AMRAD publishes a newsletter containing information on packet radio topics and other amateur radio technical pursuits, such as spread spectrum digital speech and deaf communication. Contact the above address for details on membership and newsletter.



EASTNET: A Year Later

Bob Bruninga WB4APR
59 Southgate Ave
Annapolis, MD 21401

The original goal of EASTNET to link the eastern seaboard from Washington DC to Boston was met more or less on August 6, 1984 when packets were exchanged via the repeater path WB4APR-6 in Elk Neck Maryland, WA2LQQ-0 in Warwick NY, WORLI-0 in Westford MA, and KD2S-1 in Lowell MA. Since that time numerous alternate paths have been exercised but the saturation of the primary link frequency of 145.01 MHz during prime evening hours has prevented routine end-to-end multi-hop paths.

About the time that saturation of 145.01 MHz occurred, the emergence of WORLI type bulletin boards based on the XEROX 820 system brought about a new pattern of operation which helped ease the multi-hop loading. The WORLI bulletin board systems (PBBS's) include an auto-forwarding feature which allows the PBBS's to update each other with messages destined for individuals homed on their local PBBS. By programming these PBBS's to auto-forward on the link frequencies during non prime time hours, the need for long distance multi-hop exchanges was minimized. Individuals with messages for a distant user need only post the message on the local PBBS and at the same time pick up his mail using a direct **path** or at most a single hop connection. So the result of auto-forwarding is a two fold reduction in link traffic by encouraging all messages to be posted on local paths, and forcing the multi-hop regional forwarding to occur during off hours.

The next step in relieving congestion on 145.01 is to make the PBBS's frequency agile so that they are available to local users on a local frequency such as 145.05 during prime time and move over to 145.01 only during off hours at scheduled periods for auto-forwarding. In EASTNET we have already moved the WB4APR-5 HF BRIDGE system over to 145.05 and the local PBBS of KS3Q over to 145.09. The W3IWI PBBS remains on 145.01 to serve the link traffic until frequency agility has been achieved.

HF activity is stabalizing on 10,147.900 KHz with the WB4APRO bridge into Washington, K7PYKO into Tuscon, W9TDO into Chicago, and WORLIO into Masachusetts. K7PYKO shifts between 20M and 30M as conditions dictate and WORLIO comes up when time permits.

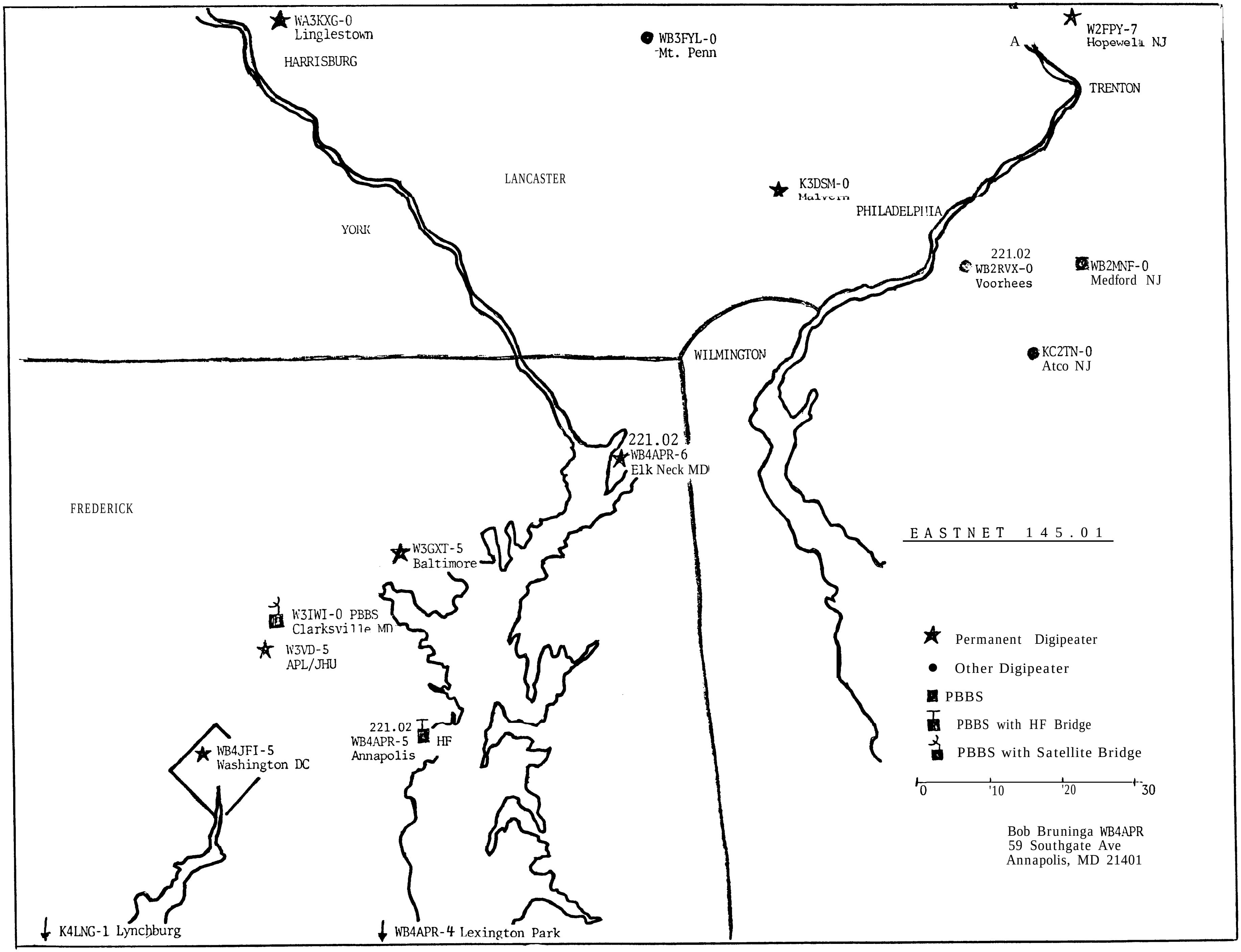
At a recent roadside gathering of enthusiasts from Virginia, Maryland, Pennsylvania, and New Jersey, the Mid-Atlantic Packet Repeater Council (MAPRC) was formed to allow a unified representation of regional packet interests. An arbitrary territory of 100 miles radius about the primary link digipeater of WB4APR6 was chosen as the area of responsibility. A similar group, the Tri-State Packet Repeater Council (TSPRC) has recently been formed to cover Northern New Jersey, New York, and Connecticut.

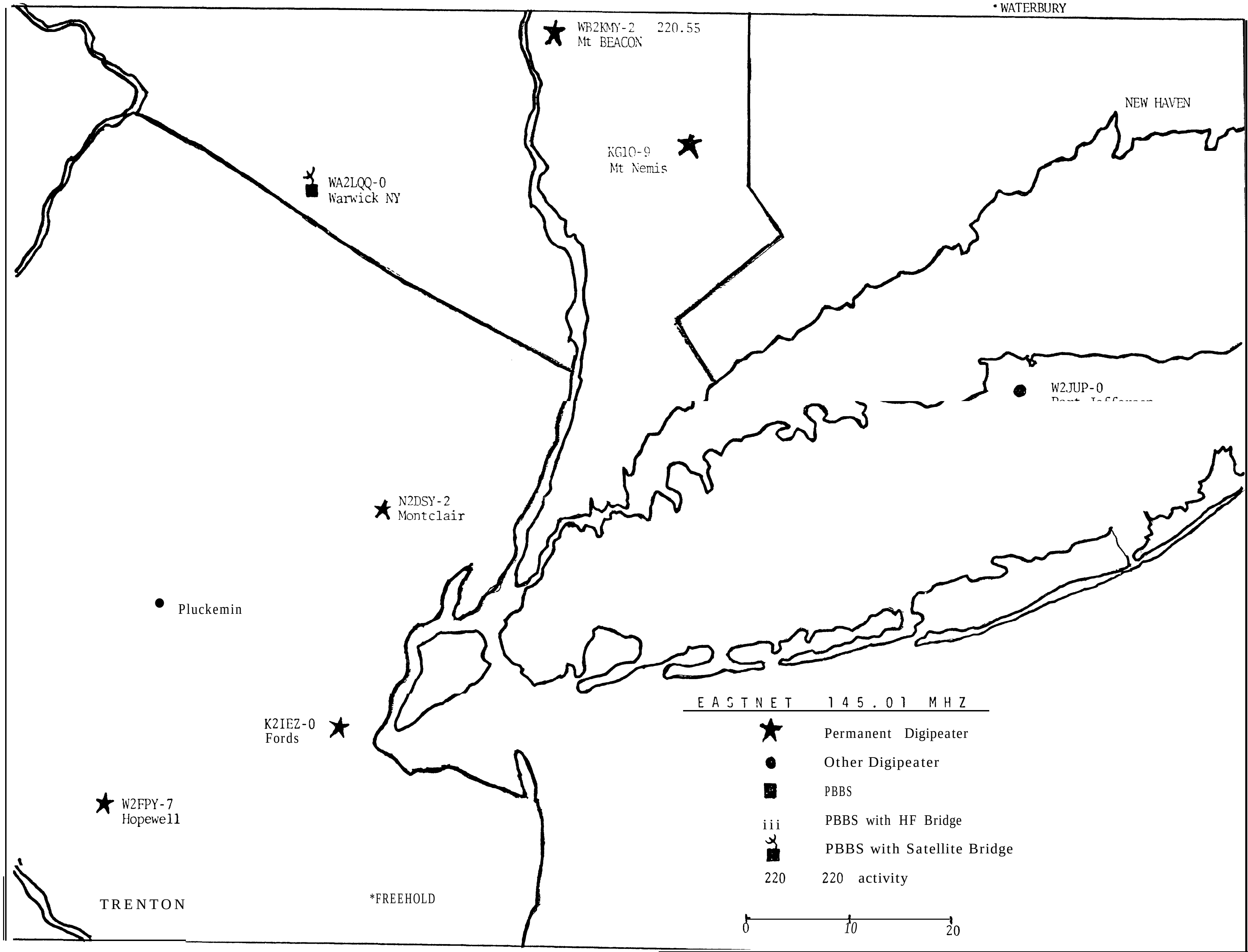
To encourage use of 220 MHz we will soon be putting a 221.02 MHz piggy-back radio on the WB4APR-6 digipeater. The 221 MHz transceiver will be connected to the same TNC but will also be under subtone control to link directly to 145.01 avoiding the digipeater for stations which are so configured. The subtone control frequency is the reverse channel tone of 367 HZ which is already built into most 202 modems. Another possibility is the use of dual port software which is just becomming available to allow a XEROX 820 system to serve both channels. Our plans still include a number of wideband channels and narrowband channels according to the following plan:

220.55		221.00	
220.65		221.02	
220.75	wideband	221.04	narrow band
220.85	100 KHz	221.06	20 KHz
220.95		221.08	

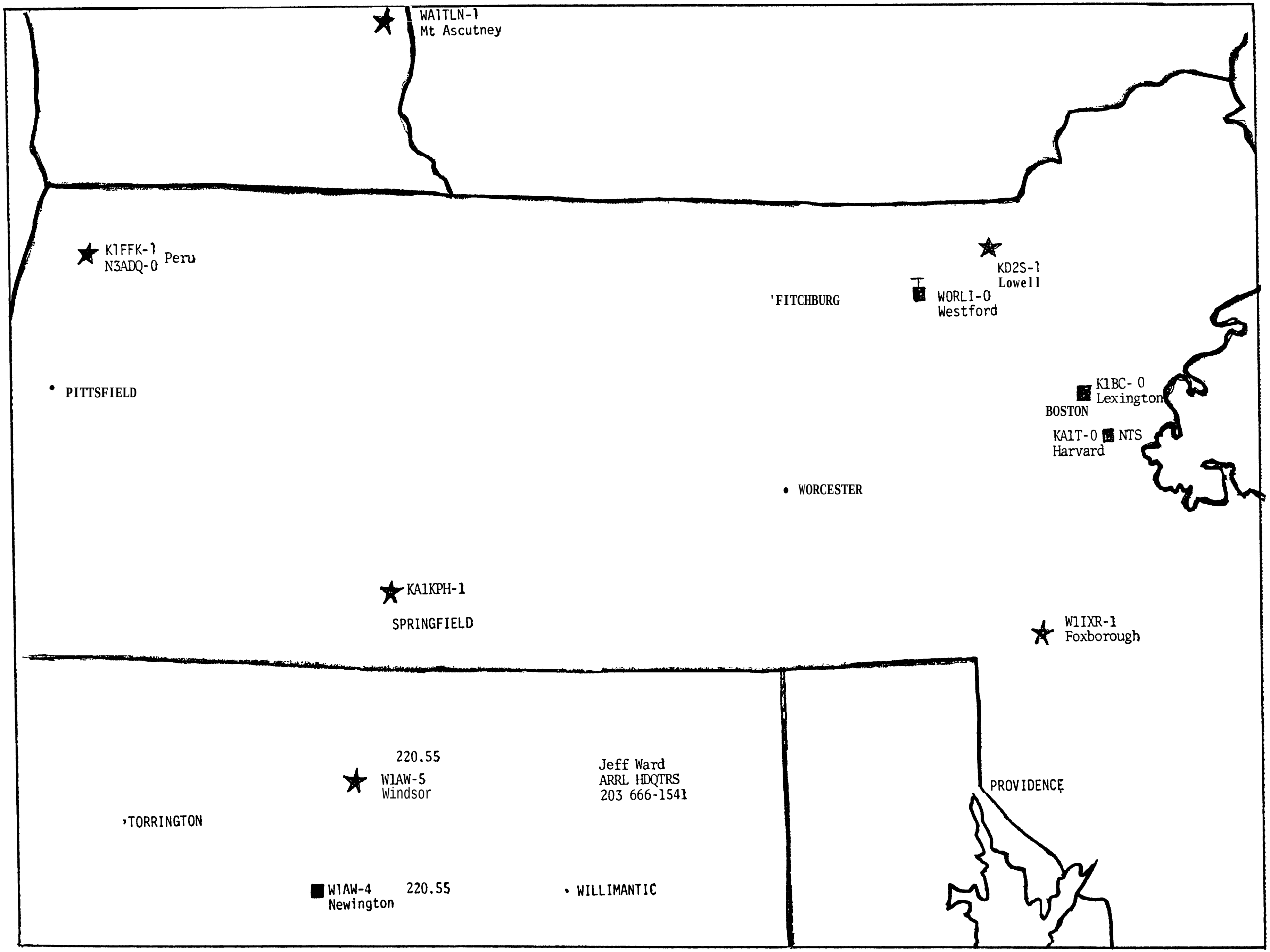
Note that this is the same as what we proposed last year except that the narrow band channels have moved down 10 KHz to match the existing 20 KHz channel spacing on the band.

On the following maps all of the known wide area use packet repeaters are indicated as well as the PBBS's and gateway stations. Indications of 220 activity should be considered to be in addition to the existing two meter operations at the same location. These maps are not expected to be 100 percent accurate and we apologize for all those who were omitted but we just wanted to give a quick overview of packet activity in various areas of the country. We will attempt to update these maps periodically and make them available next year if you will help by forwarding any permanent updates to the above address.

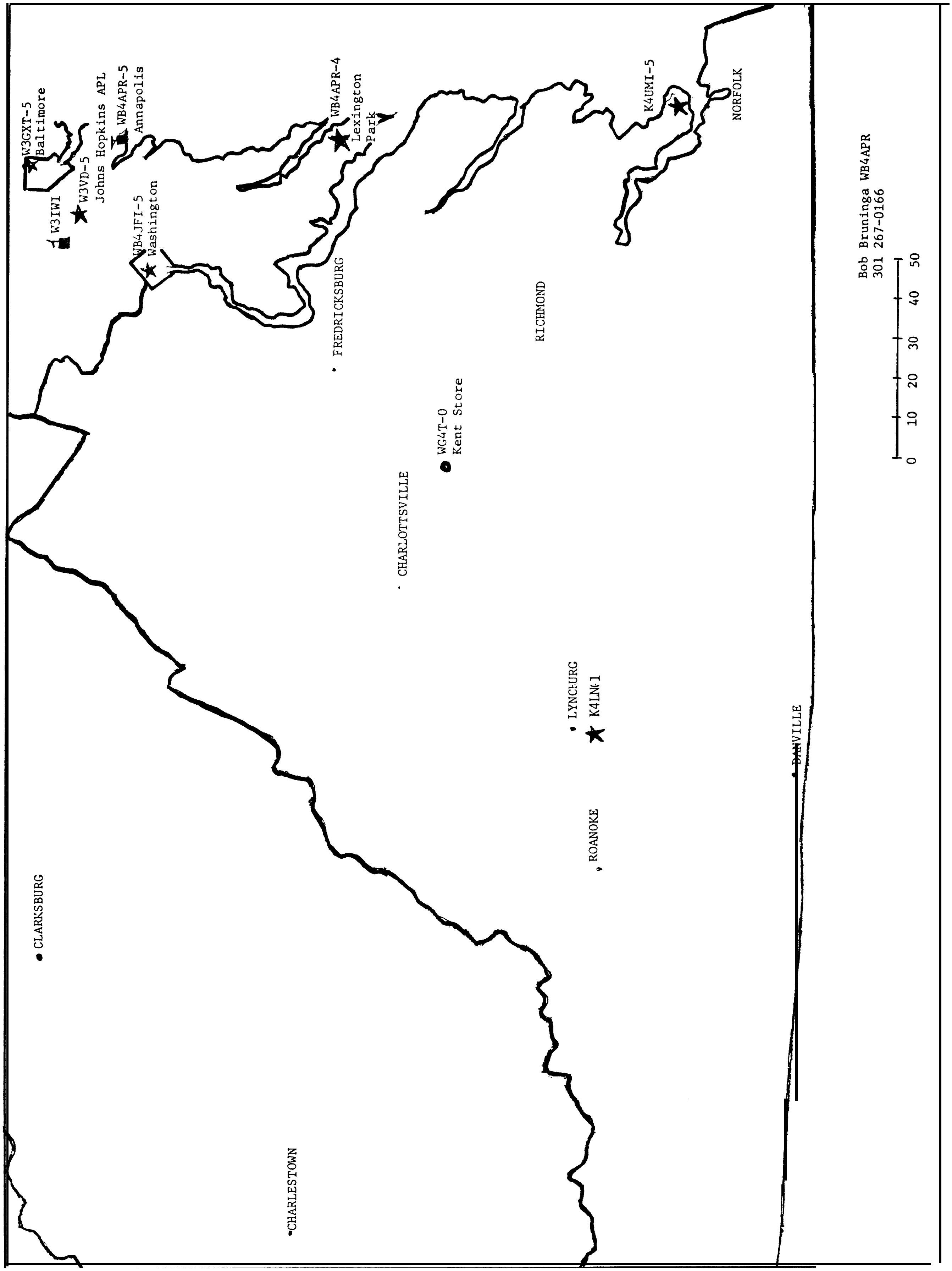

WA3KXG-0
Linglestown
HARRISBURG
WB3FYL-0
Mt. Penn
W2FPY-7
Hopewell NJ
A
TRENTON
LANCASTER
K3DSM-0
Malvern
PHILADELPHIA
YORK
221.02
WB2RVX-0
Voorhees
WB2MNF-0
Medford NJ
WILMINGTON
KC2TN-0
Atco NJ
221.02
WB4APR-6
Elk Neck MD
FREDERICK
EASTNET 145.01
W3GXT-5
Baltimore
W3IWI-0 PBBS
Clarksville MD
W3VD-5
APL/JHU
Permanent Digipeater
Other Digipeater
PBBS
PBBS with HF Bridge
PBBS with Satellite Bridge
221.02
WB4APR-5
Annapolis
HF
WB4JFI-5
Washington DC
0
10
20
30
Bob Bruninga WB4APR
59 Southgate Ave
Annapolis, MD 21401
K4LNG-1 Lynchburg
WB4APR-4 Lexington Park

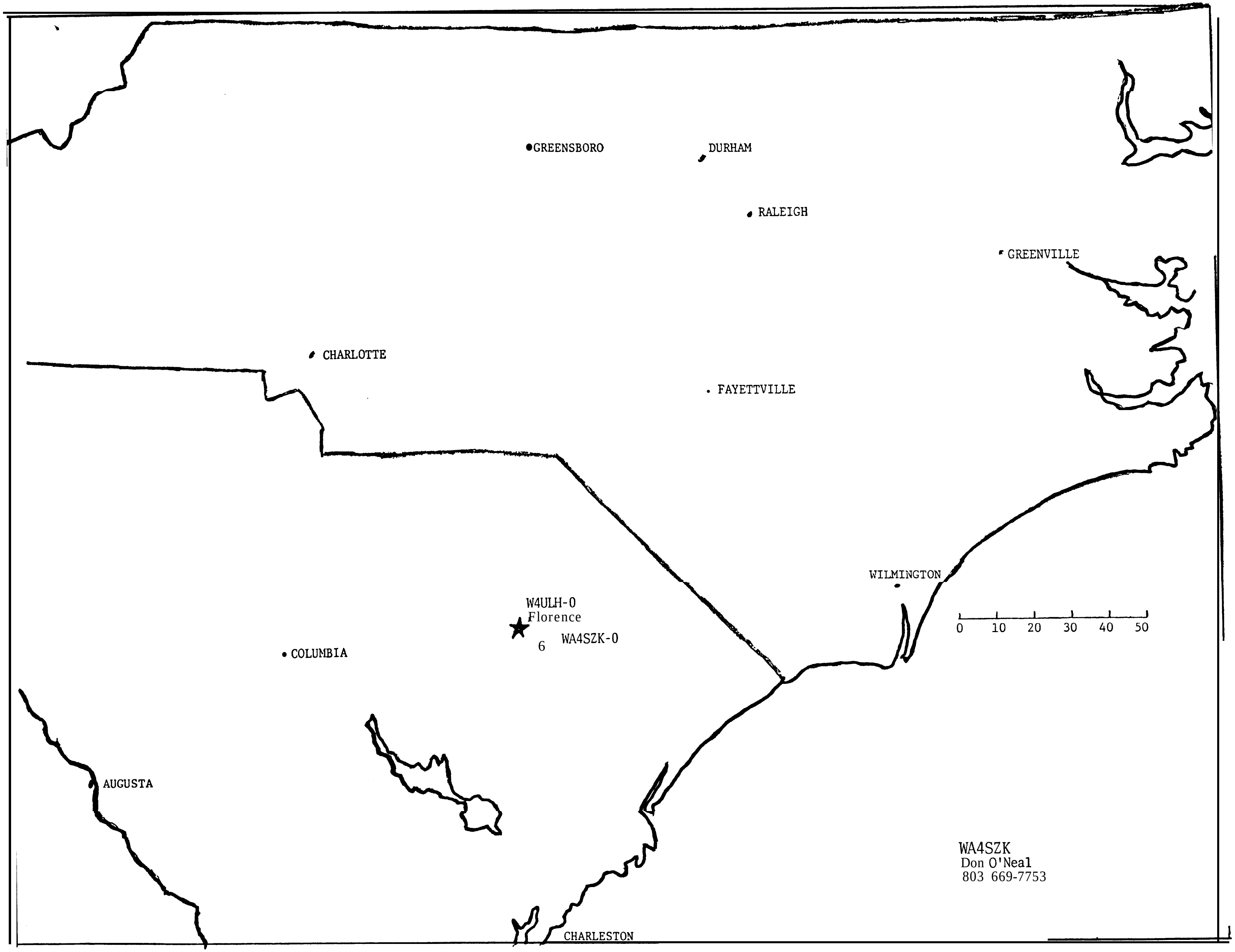


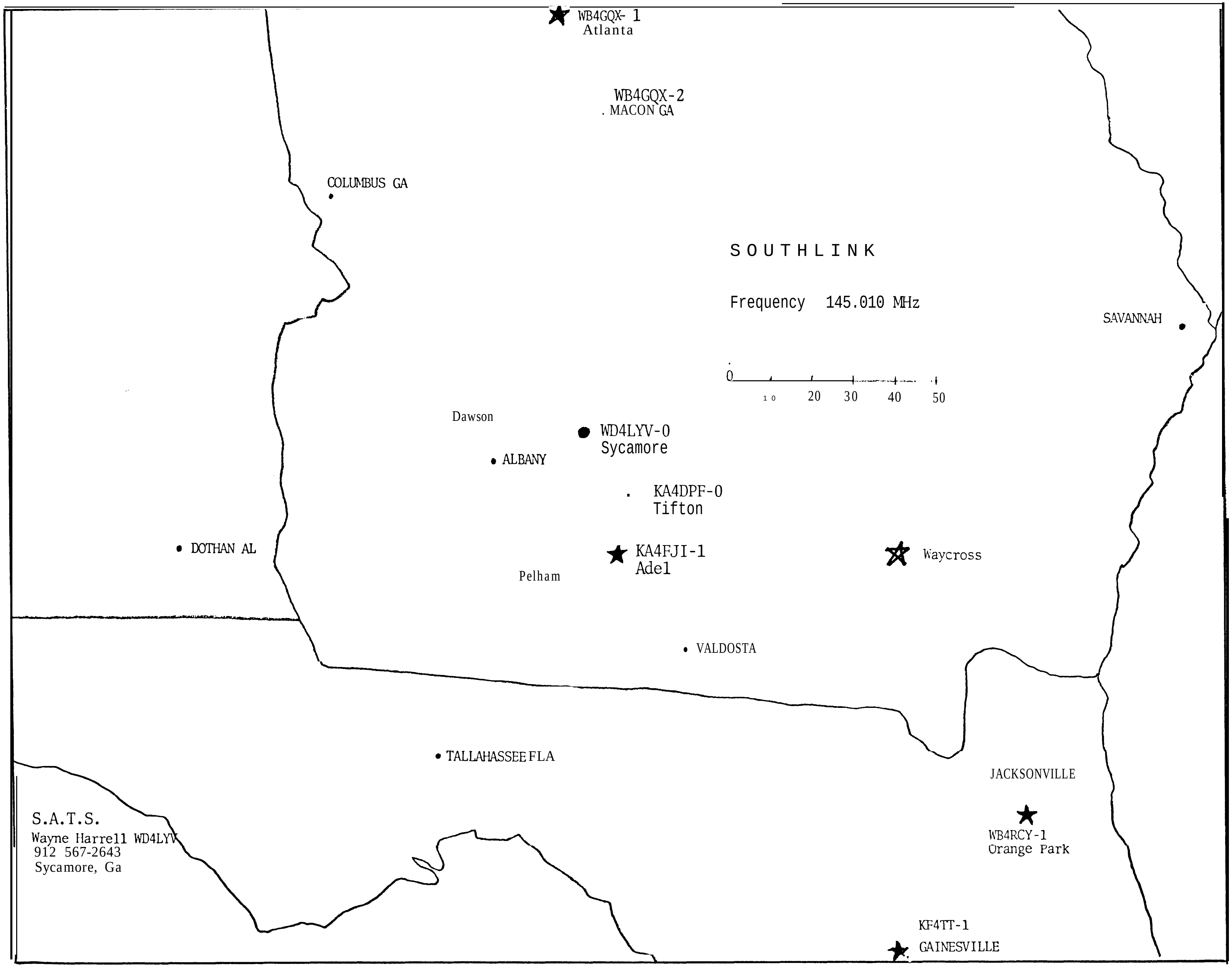
• WATERBURY
WB2KMY-2 220.55
Mt BEACON
NEW HAVEN
KG1O-9
Mt Nemis
WA2LQQ-0
Warwick NY
W2JUP-0
N2DSY-2
Montclair
Pluckemin
K2IEZ-0
Fords
W2FPY-7
Hopewell
TRENTON
*FREEHOLD
EASTNET 145.01 MHZ
Permanent Digipeater
Other Digipeater
PBBS
iii PBBS with HF Bridge
PBBS with Satellite Bridge
220 220 activity
0 10 20

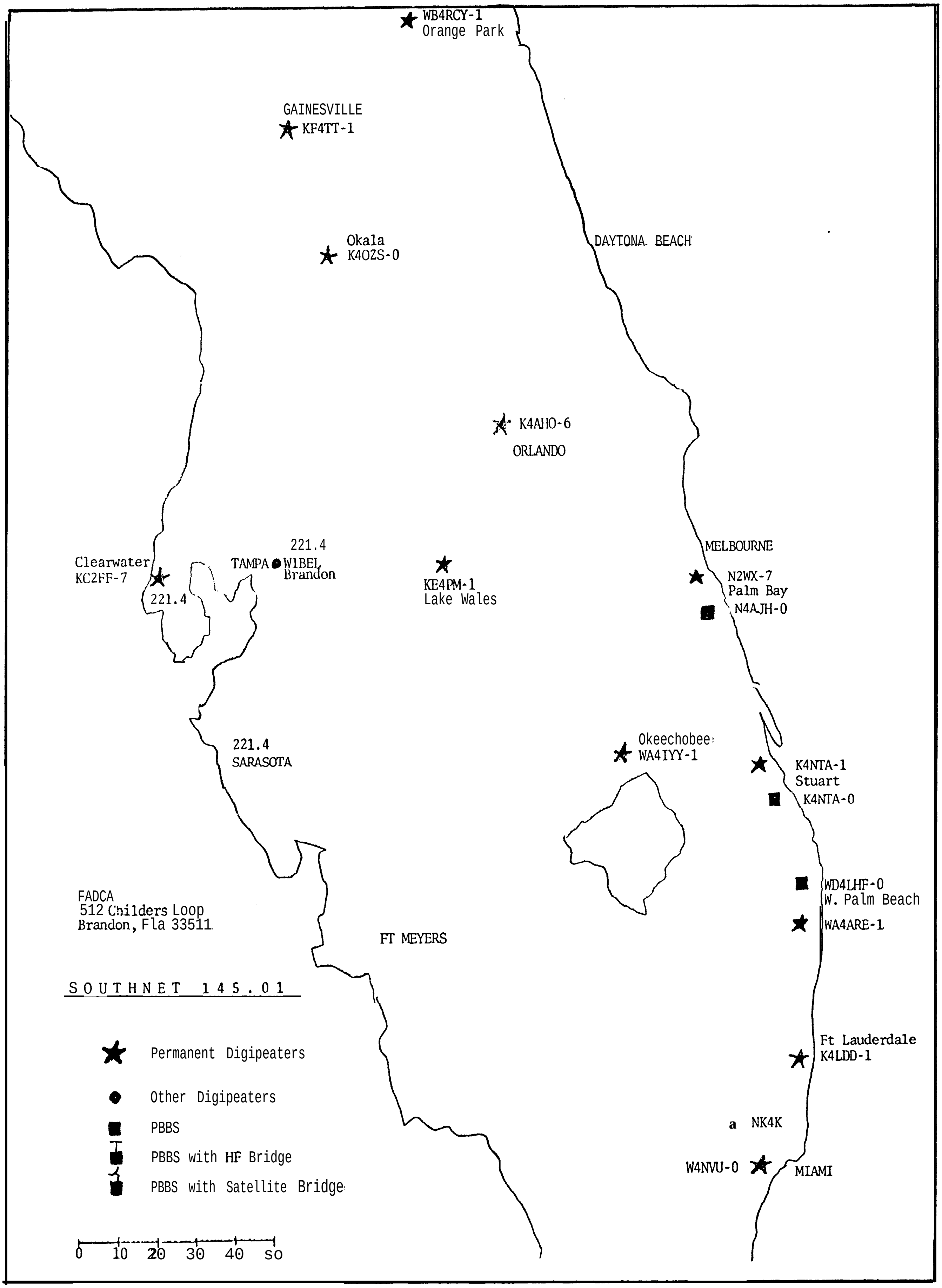


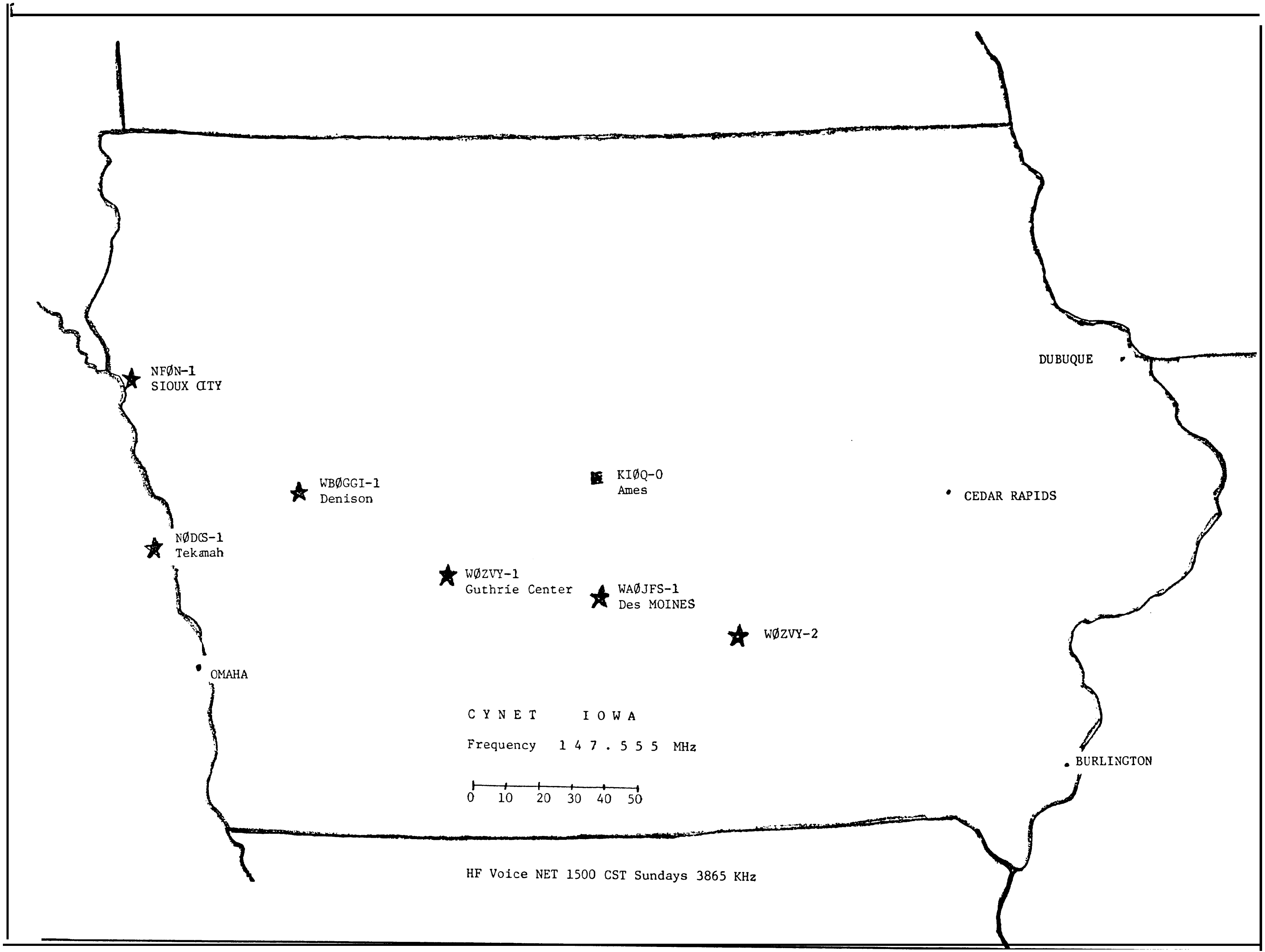
WA1TLN-1
Mt Ascutney
K1FFK-1
N3ADQ-0
Peru
KD2S-1
Lowell
FITCHBURG
WORLI-0
Westford
PITTSFIELD
K1BC- 0
Lexington
BOSTON
KA1T-0
NTS
Harvard
WORCESTER
KA1KPH-1
SPRINGFIELD
W1IXR-1
Foxborough
220.55
W1AW-5
Windsor
Jeff Ward
ARRL HDQTRS
203 666-1541
PROVIDENCE
TORRINGTON
W1AW-4
Newington
220.55
WILLIMANTIC

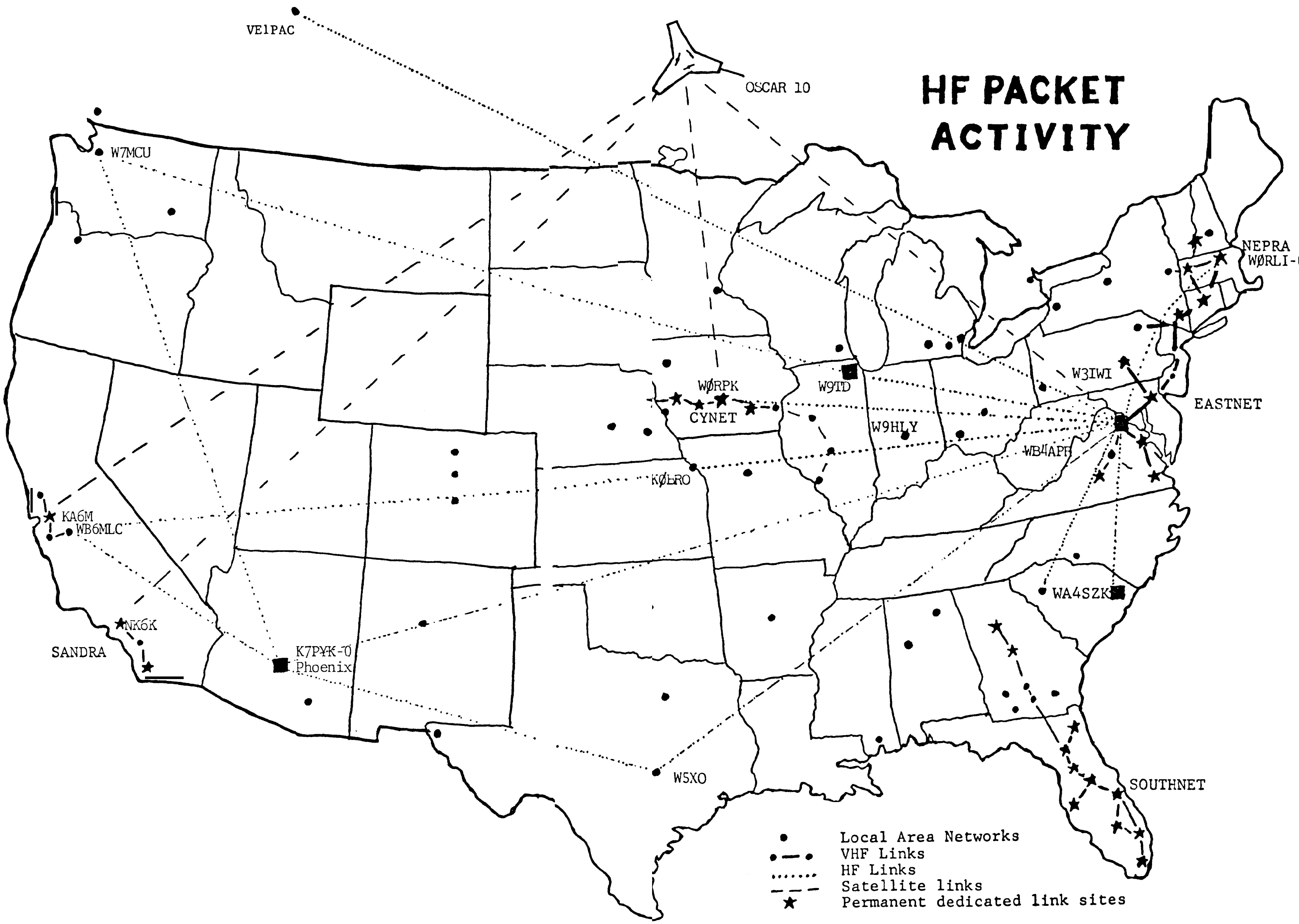

W3GXT-5
Baltimore
W3IWI
W3VD-5
Johns Hopkins APL
WB4APR-5
Annapolis
WB4JFI-5
Washington
WB4APR-4
Lexington
Park
K4UMI-5
NORFOLK
FREDRICKSBURG
RICHMOND
WG4T-0
Kent Store
CHARLOTTSVILLE
LYNCHURG
K4LNG-1
DANVILLE
ROANOKE
CLARKSBURG
CHARLESTOWN
0
10
20
30
40
50
Bob Bruninga WB4APR
301 267-0166


GREENSBORO
DURHAM
RALEIGH
GREENVILLE
CHARLOTTE
FAYETTVILLE
WILMINGTON
0 10 20 30 40 50
W4ULH-0
Florence
WA4SZK-0
6
COLUMBIA
AUGUSTA
WA4SZK
Don O'Neal
803 669-7753
CHARLESTON


WB4GQX- 1
Atlanta
WB4GQX-2
MACON GA
COLUMBUS GA
SOUTHLINK
Frequency 145.010 MHz
0
10
20
30
40
50
SAVANNAH
Dawson
WD4LYV-0
Sycamore
ALBANY
KA4DPF-0
Tifton
DOTHAN AL
KA4FJI-1
Adel
Waycross
Pelham
VALDOSTA
TALLAHASSEE FLA
JACKSONVILLE
S.A.T.S.
Wayne Harrell WD4LYV
912 567-2643
Sycamore, Ga
WB4RCY-1
Orange Park
KF4TT-1
GAINESVILLE


WB4RCY-1
Orange Park
GAINESVILLE
KF4TT-1
Okala
K4OZS-0
DAYTONA BEACH
K4AHO-6
ORLANDO
221.4
Clearwater
KC2FF-7
TAMPA
W1BEL
Brandon
221.4
KE4PM-1
Lake Wales
MELBOURNE
N2WX-7
Palm Bay
N4AJH-0
221.4
SARASOTA
Okeechobee
WA4IYY-1
K4NTA-1
Stuart
K4NTA-0
WD4LHF-0
W. Palm Beach
WA4ARE-1
FADCA
512 Childers Loop
Brandon, Fla 33511
FT MEYERS
SOUTHNET 145.01
Permanent Digipeaters
Other Digipeaters
PBBS
PBBS with HF Bridge
PBBS with Satellite Bridge
Ft Lauderdale
K4LDD-1
a NK4K
W4NVU-0
MIAMI
0 10 20 30 40 so


NFØN-1
SIOUX CITY
WBØGGI-1
Denison
KIØQ-0
Ames
CEDAR RAPIDS
DUBUQUE
NØDCS-1
Tekamah
WØZVY-1
Guthrie Center
WAØJFS-1
Des MOINES
WØZVY-2
OMAHA
BURLINGTON
C Y N E T   I O W A
Frequency 1 4 7 . 5 5 5 MHz
0 10 20 30 40 50
HF Voice NET 1500 CST Sundays 3865 KHz


HF PACKET ACTIVITY
VE1PAC
OSCAR 10
W7MCU
KA6M
WB6MLC
NK6K
SANDRA
K7PYK-0
Phoenix
WØRPK
CYNET
KØLRO
W9TD
W9HLY
WB4APR
W3IWI
EASTNET
NEPRA
WØRLI-0
WA4SZK
SOUTHNET
W5XO
Local Area Networks
VHF Links
HF Links
Satellite links
Permanent dedicated link sites

PACKET RADIO TIMING CONSIDERATIONS

David Engle, KE6ZE
1063 Summerwood Ct.
San Jose, CA 95132

ABSTRACT

This Paper presents an analysis of existing packet radio systems and equipment (2 Meter AFSK). Both dual and single frequency repeater efficiencies are analyzed. The results demonstrate the relative inefficiencies of the existing networks. These inefficiencies reduce the effective capacity of a 1200 baud channel to 300-500 baud. In order to correct some of these inefficiencies a few suggestions are offered.

INTRODUCTION

A packet radio network stat ion in switching from receive mode to transmit mode passes through several processes that consume significant periods of time. These losses are due to inefficiencies contributed by the sending station, the repeater, and the protocol utilized. In order to establish a basis for further investigation, these time consuming processes and activities are explained here:

1. Send process time -- The time the sending station's central processor (cpu) takes to assemble and dispatch the packet. This is the terminal node controller (TNC) processing time. The time consumed is approximately .250 milliseconds. This time is relatively insignificant (in proportion to the other processes) and for all practical purposes can be ignored.

2. Transmitter turn-on time -- The time the sending packet radio stat ion takes to turn on it's carrier (and/or switch to transmit from receive). This activity is often overlooked in the analysis of packet radio systems. It is the single largest piece of dead time in the packet transmission process. This time is not of any great concern in a voice system and is not generally not iced for what it is. However, it can be recognized sometimes on voice as an operator chops off his call sign announcement at the beginning of each transmission (Sound familiar? You always thought he started talking before he pushed the PTT button). The amount of time consumed by this activity can vary significantly. A transmitter altered specifically for packet radio service can come on line in several milliseconds (e.g., like CW break-in keying). Most amateurs are using commercial two meter equipment. Unfortunately, most twc meter equipment is not made for break-in keying and suffers from the following list of delays:

   a. Synthesized rigs have to allow time for the phase lock loop to acquire the transmit frequency when switching from receive to transmit. Experience has shown some rigs to require 500 milliseconds for this function.

   b. Older rigs with sol id state receivers and tube transmitters have to allow time for the transmit power supply to come on line. This act ion can take as little as 100-200 milliseconds but 500 milliseconds is more realistic.

   c. Older rigs with relay switching of the antenna can use up to 200 milliseconds to accomplish T-R switching. The time is used in circuit delays, transmit recovery and relay movement time.

   d Low powered rigs usually use a solid state in-line amplifier (brick). These units switch from receive (bypass mode) to transmit by sensing for RF energy. Upon detection of RF from the rig the PA unit switches in the high power amplifier. The time for this function to take place has been found to require about 200 milliseconds.

3. Modem turn on time -- The time a modem takes to enable and stabilize the initial tone. Most commercial units allow 50 milliseconds for this event. The time consumed by this event is usually masked by the transmitter turn on time, so it can

be ignored in most cases. However, if the system waits for clear to send (CTS) to be returned from the modem before turning on the transmitter, this unnecessary delay will be introduced. Each packet radio station should be checked to insure the modem and transmitters are turned on simultaneously. A modem constructed specifically for packet radio use can eliminate this time.

4. Propagation time -- The time it takes the radio signal traveling at the speed of light to arrive at the receiver. A nominal distance of 30 miles will take .03 milliseconds to traverse. For all practical purposes this time can be ignored.

5. Receive process time -- The time it takes the receiving station to process the packet and pass it on to the next stage (e.g., operators terminal). An allowance of .25 milliseconds should take care of this function. Once again for all practical purposes this time can be ignored.

6. Data receipt allowance -- At the end of transmitting a data packet the transmitter and modem do not turn off coincident with the last bit. Commercial modems continue to transmit the final tone for 10-20 milliseconds after the last bit. This is to allow the receiver clocking mechanism time to move out the last character received prior to the introduction of line noise that will occur with the removal of the carrier.

7. Transmitter turn off time -- Some hand held transceivers do not stop transmitting with release of the "PTT". They hang for a short time. While not a common occurrence it can prevent TNCs with carrier detect from accessing the channel. Thus, ack packets could be delayed. Reports of this happening have been seen on the UNIX USENET. This has not been seen the S.F. Bay Area and isn't included in the following analysis.

These are the inefficiencies introduced by the packet boards, modems and RF equipment into a packet radio network.

Another class of inefficiencies are introduced into the network by the AX25/HDLC protocol utilized by the network. They are:

1. Protocol overhead -- The HDLC and AX25 sync, control, address, and checksum bytes add an additional 20 or 27 bytes to each packet. They are: one sync byte (beginning), 14 or 21 bytes of address (14 for simplex, 21 for repeater), two control bytes, two checksum bytes, one sync byte (end).

2. Bit stuffing -- Bit stuffing required by the HDLC protocol requires that a "0" bit be inserted each time five "1" bits are encountered. The random probability of this occurring is 1 in 32 or 3.125%. This means that the data stream will be expanded approximately 3%. Table 1 includes transmit times for 3 packet sizes adjusted for bit stuffing.

3. Acknowledge packet -- Each message is usually acked. A receiving station sends back to the sender an ok to proceed packet of 20 or 27 bytes (20 for simplex, 27 for repeater).

The protocol itself introduces a quantity of bytes and an additional 3% of overhead (delay) into the system.

REPEATERS

The key to any packet radio system is the repeater. There are two types of repeaters in general use, single frequency and dual frequency repeaters. On a single frequency repeater (a digipeater), the repeater receives the entire packet, makes sure the packet is valid, and retransmits the packet on the same frequency. A dual frequency repeater listens on one frequency and retransmits the packet on another frequency. It does this with no delay and thus no packet validity check. Each approach has its merits and detractions. On the surface it appears as though a single frequency repeater utilizes half the bandwidth of a dual frequency repeater but takes twice as long to complete a transaction. However, the inefficiencies described above cloud the issue and the merits are not as clear as they seem.

Another point of concern in a packet radio system is the amount of data to be allowed in the packets i.e., the packet size. With a given amount of data to send a few large packets would be more efficient than many small packets. But if you have a small amount of data to send then you are wasting the capacity of a large packet. The solution appears to be to use variable size packets. Then a maximum packet size has to be agreed upon, as all members of a local packet community have to be able to receive that maximum size packet.

It should be noted that Abramson and Ferguson have theoretically shown that mixed packet sizes decreases the throughput capacity of a contention packet system. However, for better or worse the current packet radio systems allow mixed

packet sizes. As some packet systems are now running close to their capacity this effect is now being realized. Mixed packet sizes and the relative decrease in channel capacity is an effect that should be investigated.

INVESTIGATION

In order to determine the effect of these inefficiencies on packet radio transactions a few types of transact ions are examined. They are for: a simplex connection, a connection through a Dual Frequency (DF) repeater and a connection through a Single Frequency Repeater (SF). Packet sizes examined are a 64, 128, and 256 bytes of data on an Amateur radio packet system operating on a typical two meter AFSK FM carrier utilizing 1200 baud AT&T type 202 modems. The data packets conform to the AX-25 protocol. Each data packet is individually acknowledged and the ack packet is a minimum packet of 20 or 27 bytes. The times are based on the actual measured times of the KE6ZE & KA6M stat ions and the KA6M repeater. These delays are depicted in Figure 1 and Table 2.

Utilizing the data in Tables 1 and 2 the efficiencies can be computed. The results are presented in table 3 and summarized in Table 4. The efficiency is the time consumed by the actual data packet (of 64, 128, 256 Bytes) divided by the time utilized for each transaction. (Example for a single frequency repeater and 64 bytes of data: 1781.62 + 439 + 439 = 2659.62, 439 / 2659.62 = .16506 or 17%. Have to add two 64 byte transmission times because the data was sent twice, once to repeater, then repeater to destination (see Figure 2). Note: Table 3 accounts for the other multiple hop times.) Notice the the two dual frequency efficiencies. The first is calculated on an elapsed tine basis, the second on a spectral density time basis. This is to relate the two repeater types on an equal basis. i.e.: if a single frequency transmitter utilized twice the data rate and thus twice the bandwidth then the two would be equal in spectral density.

Note the decreased efficiency of the smaller packets. This is one case where bigger is better. Also, both repeater types are relatively inefficient at the small packet sizes utilized in most packet radio systems at the present. It appears as though we have considerable room for improvement in our systems. Note the advantage a single frequency repeater system offers in its ability to allow users to direct connect. At 25% efficiency a 1200 baud system will have the throughput rate of 300 baud.

CURES AND FIXES

What can be done to reduce some of these inefficiencies? One thing is to improve the data stream transmitted over the ether. A single frequency repeater can offer efficient use direct connects to its users, while having the repeater available should it be required. Make bigger packets where possible. But not too big, there will be both rag chewing traffic and computer traffic, each at opposite ends of the packet size needs. Attention to modem timing can increase efficiency also.

Another opportunity is to improve the rig for packet radio. The choice is to either make or buy a rig, but believe it or not the best may be to buy a used rig. There are old mobile FM voice rigs that are suitable for 2 meter packet radio service. They are generally available used at reduced prices and should have the following characteristics:

1. Crystal controlled receiver with another crystal for control of the transmitter frequency.

2. All solid state unit with an integrated single power supply. Although some hybrid rigs with solid state receivers and tube transmitters are more likely to be found. These have been found to come on in about 200-300 milliseconds. An all sol id state can be cut down to less than 100 milliseconds for turn on.

3. Moderate power RF amplifier contained within the rig. i.e., try to keep from using an external amplifier.

4. Antenna switching handled by a reed relay or PIN diode blocking (PIN diode is the best of the two).

Fortunately enough there are older (obsolete ?) rigs with most of these characteristics. They are crystal controlled mobile rigs. Progress in technology for voice rigs has relegated these units to the storage shelf and/or flea market sales. If you are looking to buy a rig and devote it to packet radio at your next flea market look for an all sol id state, crystal, mobile rig. For packet radio you rarely need more than several frequencies. So, the limited channel selection should not be a disadvantage. They are small and can be easy to shoe horn into a mountain top site. However, they have moderate power output (10 watts or so>. If they can be used bare-foot they make a fine packet station.

There are also lots of handy-talky (HT) units available at flea markets. Generally HTs have disadvantages of not having enough power and/or being synthesized. If your repeater or stat ion is operating simplex then the PLL lock time may not effect transmitter turn on time.

It depends on whether or not the unit always waits for a time to lock or senses. PLL lock. The lock time in addition to an outboard linear switching time was enough to require the first KA6M repeater 500 milliseconds for transmitter turn on time. The temptation to make a small packet station using an HT is great. If you build a repeater (or station) with a low powered HT or mobile rig and a linear, be sure to key the linear from the TNC PTT. Don't use RF sensing to switch the linear.

Also available at Flea Markets are older commercial mobile rigs (e.g., Motorola Motrac). These hybrid (solid state receiver, tube transmitter) rigs can be utilized in packet service if one gives consideration to the power supply turn on time required by them. Careful attention and adjustment can get them down to 150-250 millisecond turn on time. They have the advantage of 25-50 watts of power and good reliability. The lack of more than a dual frequency control may hamper versatility at home but for a bulletin board system they may be ideal.

CONCLUSION

A packet radio system has delay introduced by all its component parts: the originator, the repeater, and the recipient. The key item is the repeater. The repeater should have the most attention given to it in elimination of dead time. Practices incorporated into repeaters as a result of voice techniques should be re-examined. As the transmissions are not intended for the human ear the repeater can be tailored for efficiency. These tailorings are:

1. Eliminate the squelch tail on a two frequency repeater. Holding the carrier on may keep other transmitters from utilizing that time.
2. Key the transmitter on a two frequency unit from a tone detection on the receiver side. This will keep intermodulation product interference down and also drive the kerchunkers crazy.
3. In a two frequency repeater key the transmitter immediately upon tone detection. Or upon carrier detection if tone detection is not used. Some voice repeaters require a signal be present for a short time prior to keying the transmitter. For digital use this should be eliminated.
4. Strive to incorporate as many as possible of the previously described transmitter features into the repeater.
5. Turn the squelch up on the receive system. The data will be lost on a noisy signal anyway.
6. Have the CW IDer use one of the modem tone frequencies. This will allow channel busy detection schemes using tone detect rather than carrier detect from having data collisions.
7. Have the CW IDer come on at the end of a transmission in a two frequency system. This will keep from clobbering a data packet transmission on a start up after 5 minutes has elapsed.

Paying attention to the repeater can reap significant benefits to all the participants in a packet radio system. Elimination of a unit of time from the repeater will save two units of time on a transaction: once on the data packet, and again on the acknowledgement packet.

Round up an oscilloscope and check your packet radio system. After gathering measurements on your system run through a calculation set as done here. Doing this you will probably find out what it is that is causing your 1200 baud system to act like a 300-400 baud system. Don't think that other digital communication schemes are much better at efficiencies either. AMTOR Baudot RTTY communication starts off at 63% efficiency (5 data bits and 3 overhead bits for each element) then the AMTOR overhead starts wearing that down.

CLOSING COMMENTS

Finally, in closing, there are numerous theoretical studies of packet radio systems in various technical journals. These studies in general do not account for all of the delay types listed in this paper. When analyzing your packet radio system by these studies don't ignore these practical effects. In particular, transmitter turn-on time can cause difficulty in carrier detect protocols such as utilized on the VADGC and TAPR boards. The value of the carrier detect is somewhat diminished by the time it takes to turn on a carrier e.g., after a station senses the medium for carriers and prior to turning on its carrier a significant amount of time can pass, during which another (undetected) carrier may appear. Analyses of system stability, bandwidth capacity and packet delay should take this factor into account.

64 Bytes	=	439 milliseconds
128 Bytes	=	879 milliseconds
256 Bytes	=	1758 milliseconds

Table 1
Time to Transmit Data
Includes 3% Bit Stuffing

Simplex Direct Connect	Dual Freq Repeater A	Single Freq Repeater	
.25	.25	.25	Transmit Process Time
-		.25	Repeater Process Time
-			Modem Turn On Time (masked by next item)
250	250	250	Transmitter Turn On Time
-	500	250	Repeater Transmitter Turn On Time
133	133	180	AX25 Overhead (20/20/27 bytes @ 1200 baud)
-		180	Repeater AX25 Overhead (27 bytes again)
-			Data Transmission (see table 2)
-			Data Bit Stuffing 3% (see table 2)
15	15	15	Data Receipt Allowance (carrier hang on)
-		15	Data Receipt Allowance for Repeater
.03	.03	.03	Propagation Delay
-	.03	.03	Propagation Delay for Repeater
.25	.25	.25	Receiver Processing Time
398.53	898.56	890.81 =	Time to transmit an empty (or control) packet from source to destination.
x2	x2	x2	
797.06	1797.12	1781.62 =	Time to transmit an empty packet (source to destination) and ack it (destination to source)

Table 2
Times of the component parts of packet transmissions.
Times in Milliseconds

Simplex, Direct Connect

64	128	256	
1236	2555	1676	= Milliseconds
36%	69%	52%	= Efficiency

Dual Frequency (Voice) Repeater

64	128	256	= Bytes of data
2236	2676	3555	= Milliseconds
20%	33%	49%	= Efficiency

Single Frequency (Digital) Repeater

64	128	256	= Bytes of data
2660	3540	5278	= Milliseconds
17%	25%	33%	= Efficiency

Table 3
Complete Transactions of Varying Packet Sizes
Times In Milliseconds

64	128	256	= Packet size
17%	25%	33%	= Single frequency efficiency
20%	33%	49%	= Dual frequency efficiency
10%	16%	25%	= Dual frequency spectral efficiency
36%	52%	69%	= No repeater direct connect efficiency

Table 4
Channel Data Rate Effective Utilization Efficiency

```
I-I--+---------------I---+------------+---I-------I--I--I
 | |  |                |   |            |   |       |  |
 | |  -transmitter     |   -data        |   |       |  -receive process
 | |    turn on time  -AX25 header      |   |       -propagation
 | -modem turn on time                  |   -data receipt allowance
 -transmit process time                 -AX25 trailer
```

Figure 1
Time Delay Accumulations

```
 Hop 1          Hop 2         Hop 3          Hop 4
+-------------+-------------+-------------+-------------+
 |             |             |             |
 -data packet  -data packet  -ack packet   -ack packet
  src to rpt    rpt to dest   dest to rpt   rpt to src
```

Figure 2
Dual Frequency Repeater Packet Transmissions

REFERENCES

Abramson, Norman. “The throughput of packet broadcasting channels,” IEEE Transactions on Communications, corn 25:1, January 1977

Borden, David W. “On the use of a two-frequency traditional voice repeater for local area packet networking,” Volume 2, Proceedings of the the First ARRL Amateur Radio Computer Networking Conference, 1981.

Ferguson, M. “An approximate analysis of delay for fixed and variable length packets in an unslotted Aloha channel,” IEEE Transactions on Communications, corn 25:7 July 1977.

Kleinrock, Leonard, Queuing Systems, Volume II: Computer Applications. New York: Wiley Interscience, 1976

Kleinrock, L. and F. Tobagi. “Carrier sense multiple access for packet switched radio channels,” Proceedings of the International Conference of Communications, June 1974.

Kleinrock, L. and F. Tobagi. “Packet switching in radio channels: Part I-Carrier sense multiple-access modes and their throughput-delay characteristics,” IEEE Transactions on Communications, com-23:12 December 1975

Magnuski, Hank “On the care and feeding of your packet repeater,” Volume 2, Proceedings of the First ARRL Amateur Radio Computer Networking Conference, 1981.

Tobagi, F. and L. Kleinrock. ‘Packet switching in radio channels: Part II-The hidden terminal problem in carrier-sense multiple-access and the busy tone solution,” IEEE Transactions on Communications, com-23:12 December 1975

OF VIRTUAL CIRCUITS, DATAGRAMS, AND THE CIRCULAR FILE

Terry Fox, WB4JFI
President, AMRAD
1819 Anderson Rd.
Falls Church, VA 22043

Introduction

This paper presents a slightly biased view of the main two types of networking concepts being discussed for amateur radio.

Overview

Amateur packet radio made a major breakthrough last year. After a couple years of development, a standard has been adopted for point-to-point packet communications, often referred to as the Link Layer, or Level 2 of the IS0 reference model.

Even as work was being completed on the link layer, amateurs were beginning to take on the challange of designing a true amateur packet network system. Two "camps" have taken shape in this stage of development work, the 'Virtual Circuit' camp and the Datagram' or 'TCP/IP' camp. Both groups are working on software, and I believe both will be used for a period of time to see which is best suited for amateur packet radio.

One thing both groups generally agree on is that what must be provided by the amateur network is a method of getting data from a source to a destination fairly reliably. Both groups agree that this should be assured by a transportation device at each end of a communication path, and that this communications path be absolutely reliable if necessary. This means both parties are actually designing systems that function at both levels 3 and 4 (network and transport layers). The result of this work should create "virtual-connections" between two interconnected devices within the amateur network. This virtual-connection exists between the involved devices at the interface between the Transport Layer of the IS0 reference model and whatever layer resides above it (such as a Session Layer). Since some may object to the term "virtual connection", I will instead use the term "logical network connection".

Unfortunately, the word "network" has come to mean many different things. It can mean the general concept of a large group of nodes interconnected so that data can flow back and forth between any nodes within the group. This type of network can be geographically small (as in Local-Area-Network, or LAN) or large (such as the Telenet Network). This size grouping can add to the confusion when discussing networks.

The term "network" can also mean the specific "Network Layer", or Level 3 of the IS0 reference model. The network layer is sometimes considered two sub-layers, which can also be confusing.

Throughout this paper, I will use the term "amateur network" when discussing the overall network concept. I will use the term "transport entity" when describing the interface between the upper IS0 layers and the amateur network access point. When discussing a single cluster of potentially interconnected stations (such as a group of VHF packet stations within communications range), I will use either the term "intranet" (thanks Paul!), or subnetwork, as the IS0 calls it. The term "internet" (note lower case) will be used to describe the potential interconnection of individual intranetworks to form an amateur network. This is different from Internet, which is a specific internetworking protocol.

Services Renderred By The Amateur Network

In the most basic terms, the amateur network should provide a means of transferring data from one amateur to another amateur. Ideally, both data integrity and transfer speed are important to all amateurs, but integrity and/or speed may be compromised in individual situations. The amateur network should be flexible enough to handle such special requests as reduced integrity to increase throughput (speed) for applications such as packetized voice. The other end of the pendulum is equally important. If an amateur wants to send a machine language program across the amateur network, speed may be sacrificed in order to insure absolute data integrity.

Since we amateurs live in the real world, and amateur radio is our hobby (it doesn't feel like it sometimes though), it is important to realize that whatever we do is on a small budget, and will likely suffer some disaster eventually. The amateur network should be designed with this in mind, and should be resiliant enough to cope with parts of it going down from time to time. Whenever possible, the amateur network should recover from difficulties without the users of the amateur network knowing something happened.

If a user of the amateur network knows what path through the amateur network is used to establish a network interconnection to the amateur he/she wishes to communicate, the amateur network should attempt the network interconnection in that manner. If, on the other hand, tha amateur doesn't know the path to the other amateur (or even the destination transport entity where the other station exists), the amateur network ideally should provide some type of directory to aid in establishing the network interconnection. Obviously, this directory is a frill that won't be around for a while, but some method of using it should be provided.

Sometimes it may be advantageous to provide some method of allowing the amateur network (or the other amateur's station) to directly read the status of, or control some parameters of an amateur's packet system. This may allow the amateur network to optimize level 2, 3, and 4 timers, control viewing of passwords, etc. This is sometimes referred to as an alternate control path to the amateur's packet system.

The amateur network should also allow some method of network management by requesting the status of the amateur ntework, along with controlling certain functions of the amateur network. This should be done in various levels of control, along with having geographical boundries. Traditionally, amateurs prefer to operate in a non-autocratic enviroment, so a single amateur network control group is probably beyond possibility. A hierarchical system of control would be called for, allowing some amateurs to manage their local intranet, while others would manage a larger part of the amateur network.

Cutting Up The Amateur Network Pie

This amateur network is not going to blossom overnite. It will probably take much longer to develop than the level 2 standard did. Part of this is due to the added complexity of having multiply-interconnected devices that are so interdependant on each other. In order to speed up amateur network development, along with conforming to the IS0 reference model, the amateur network should be broken up into several parts,

each of which is responsible for a portion of amateur network operation.

Transport Layer Services and Responsibilities

The Transport Layer (OSI Level 4) provides a method of transferring data transparently through the amateur network between Session Layer entities such that the session-entities don't need to be concerned about assuring reliability or speed of data transfer through the amateur network.

The Transport Layer does this by using an end-to-end protocol between the Transport devices at each end of a network interconnection. This protocol is responsible for establishing a network interconnection between two amateurs; maintaining data integrity, proper data sequencing, end-to-end flow control, and end-to-end error recovery during data transfer between the amateurs; and the release of the network interconnection when it is no longer needed. It should be noted that some of these functions may be altered/removed if requested.

The Transport Layer is relieved of routing, relaying, and non-end-to-end flow control decisions by the network layer operating underneath it.

The complexity of the Transport Layer is very dependant on the type of network operating underneath it. Some network protocols require a large Transport protocol to correct for potential problems, while other network protocols require almost no transport protocol.

Network Layer Services and Responsibilities

The ISO defines two portions of the Network Layer. Subnetworks are of one or more intermediate systems which provide relaying of data through which end-systems may establish network-connections. A Network is considered the interconnection of these subnetworks to provide a communications path between Network end-points.

The Network Layer (Level 3) is responsible for establishing a data path between two Transport Layer entities wishing to communicate through the amateur network. The Network Layer should provide this service to the transport layer in such a way as to make invisible how the network routed the data. This includes how many hops or relays it took, how many subnetworks it went through, and how many data links were used. As such, the service provided at each end of a network-connection should be the same, even if dissimiliar subnetworks are used somewhere between the two end-points.

The quality of service provided is negotiated between the transport-entities and the network-entities at the time of network-connection establishment. If a quality of service is agreed to, that quality of service shall remain in effect throughout the lifetime of a network-connection.

The Network Layer provides the following functions:

a) routing and relaying;
b) network-connections;
c) network-connection multiplexing
d) segmenting and blocking;
e) error detection and recovery;
f) sequencing;
g) local flow control;
h) expedited data transfer;
i) service selection; and
j) Network Layer management.

The Network Layer data is transferred between individual network-entities through the use of Level 2 connections. In the amateur network, this usually means AX.25 HDLC connections between network nodes or entities. Level 2 AX.25 is responsible for providing reliable node-to-node data paths between the network nodes.

An important point is that the quality of service provided by the overall amateur network is only as good as the weakest portion of the path through the network.

Proposed Datagram Network Standard

The datagram network crowd is proposing the use of the DARPA TCP/IP or UDP/IP standards in building the amateur network. The Internet (IP) protocol would be used at the network layer, and either the User Datagram Protocol (UDP) for use in unsophisticated transport enviroments, or the Transmission Control Protocol (TCP) for more reliable transport service.

Proposed Virtual-Circuit Standard

Most of the work being done in the virtual-circuit area is being based on CCITT standards. One recommendation being proposed is as follows:

Use CCITT X.25 Level 3 protocols for the connections between amateur network users and the amateur network entry point.

Use the CCITT X.75 Level 3 protocol for the connections between devices within the amateur network.

Use the CCITT X.224 Level4 protocol for the Transport connections (if necessary) between the two end-points of the amateur network.

Head-To-Head Comarisons Of Virtual Circuits And Datagram Type Network Operation

As will soon become apparent, both the virtual-circuit and datagram network concepts have good points and bad points. It will be up to the amateur community and network designers to decide how these will be used in differing operating enviroments.

Both of the amateur network concepts will create a logical network connection between the two end-points of the amateur network wishing to communicate. Both will have the capability of providing either reliable data transfer, or reduced reliability in favor of increased speed of data transfer.

Design Philosophy

Even though both network designs provide the end users the same service (potentially error-free data transmission from source to destination), the way the two systems accomplish this goal is quite different.

The datagram type network design works much like the way mail is delivered by the post office. Each letter (packet) has all the information necessary for that letter to be delivered independently of any other letter before or after it. Each datagram packet has both the source and destination addresses in a header prepended to the user data, along with some control information. This packet is then shot out into the air independently of how other packets for the same source were sent. It is up to the Transport Layer to make sure the packets do get from source to the destination in the proper sequence and without corruption. This means that in a datagram network, the Transport Layer is relied on heavily to correct for Network Layer problems.

The virtual-circuit type network operates more like the telephone system. When a telephone user wishes to talk to another telephone user, the first user establishes what looks like a direct wire circuit between the two users by dialing the destination users number. Once the call is established, every word (packet) flows from the source to the destination over the same circuit. Since the same circuits are used throughout the connection, it is not necessary to have an overseeing device make sure the wires don't move or change during the connection (yes, I realize there is multiplexing and line switching going on these days, but lets not confuse the issue). When the users are done, one hangs up, and that triggers the tearing down of the circuit, making the wire connections available to others.

It is now time to discuss some of the trade-offs between the two types of systems.

Packet Header Overhead

There is a large discrepency in the amount of header type overhead that the two network designs require. This may or may mot be important, but should be considered.

In the datagram network, a minimum of 20 bytes of overhead is required by the Internet Protocol, with an additional amount required if options are to be selected. The Transport Layer protocol (TCP) requires an additional 20 bytes minimum, again more is required if options are selected. Keep im mind that this 40 bytes minimum is required in EVERY SINGLE data packet sent.

The virtual circuit network proposed relies on the fact that all the addressing information is loaded up in the connection establishment process. This can be up to 256 bytes of data in the first connection request packet (assuming the Transport Layer connection request is in the fast-select portion of the network connection request). Once the connection is made, and as long as no major errors occur, the overhead drops drastically to three bytes for the Network Layer header and three to nine bytes of Transport Layer header overhead per data packet.

It looks like the virtual circuit network design wins this one hands down.

Packet Resequencing

In datagram type networks, it is possible for packets sent after others to arrive at the destination before the earlier sent ones. This is similar to when two people correspond every day through the mail, sometimes a letter sent after another arrives before the earlier sent one. There MUST be some method of making sure that the out-of-sequence packets are re-sequenced before they can be delivered. While I have heard and read that some people consider this a "trivial" task, it does take up buffer space and processor time at the destination end-point.

Since virtual-connection networks always use the same path for every packet (unless there has been a malfunction), the chances of this problem occuring are virtually eliminated, reducing processor and memory requirements.

Once again, the virtual connection protocol seems to have the advantage.

Routing Selection

If the route through the amateur network is static (not altering for any reason other than network device failure), it can be argued that both types of network designs work equally well. The selection of routes for packets is in itself another argument for another time. It can also be argued that in a fully static network, the virtual connection may have a slight advantage, since the address overhead is not required if no decisions are to be made based on these addresses.

If dynamic routing is allowed (where changes in the route of packets from source to destination can occur for a variety of reasons), the datagram type network has a distinct advantage. Since each datagram contains both address, routing decisions can easily be made, in worst case on a packet-by-packet basis. Since the virtual connection reduces its overhead by sending the addresses only during the connection establishment process and uses "logical channel numbers" from then on, it cannot easily alter the path of packets. Keep in mind that dynamic routing may add more problems than it corrects. Network oscillation, delays due to routing decision time, and sequence destruction are but a few of the problems associated with dynamic routing.

Congestion Bypassing

Avoiding routes that have become congested is only viable when some form of dynamic packet routing is employed. Since virtual connections do not lend themselves to dynamic routing of any kind, the capability of bypassing areas of congestion is a definate advantage of the datagram form of network. The only method of reducing congestion problems in virtual connection networks is to provide some sort of look-ahead routine to make sure that congestion is cut-off before it becomes a problem. Admitedly, this is a poor form of dealing with this situation. The datagram becomes the big winner here.

Tolerance to Switch Failure

There are two issues to be concerned with in talking about packet switch failures. The first is what happens to the rest of the network when a switch fails, and the other issue is how does the switch itself recover from a failure (even a temporary one such as a power glitch). It appears that the datagram network is more resiliant in both these issues. If a packet switch fails in a virtual connection network, all connections through that switch must be torn down and re-established using another path (if available). The datagram network may have to do a similar process if it is totally static routed, but if some form of dynamic routing is used, recovery is made much easier by just re-routing the data around the failed switch.

The other issue is that of switch recovery. When a packet switch has recovered from a failure in datagram network, it just has to rebuild its routing table and inform the network it is back in operation. The virtual connection switch must do this plus re-initialize all the connections passing through it. An additional problem is that some virtual connections may not realize that the switch has failed, causing additional hardship for the switch.

It appears that the datagram network is ahead on this one also. Measures such as battery backup and uninterruptable supplies can help to reduce this, but again this is a kludge.

Reliability/Speed Tradeoffs

Much has been made of this by the datagram group. It appears that even though both networks can be made to allow for reduced reliability in order to improve speed when the reduced reliability isn't a concern (such as packetized voice), the datagram network won't try to force the reliability issue like the virtual connection network would. It is up to the reader to decide if this is a real or imaginary advantage. It appears to be much easier to make a solid pipe (virtual connection) leaky by poking holes into it than to try to plug up the holes of a leaky pipe (datagram). At this point in time, I think this is almost a non-issue.

Roving Station Situation

It isn't much of a problem at the moment, but some thought should be given to the concept of a mobile packet station, either in an auto or an airplane for example. First thoughts seem to indicate that datagrams have an advantage in this situation. This is NOT so. Since both network designs rely on providing a logical connection through the amateur network from a source end-point to a destination end-point, if one of these end-points was to change, both types of networks would have to re-establish the connection to the new end-point. It may be argued that datagrams may be easier to do this, since a whole connection doesn't have to be torn down and a new one errected. Since the Transport Layer devices must be changed anyway, the form of network re-establishment is not a major issue. Both forms of networks could employ similar methods of causing this reconnection to happen.

Alternate Data Path

Sometimes it is advantageous for either the network or the remote end user might want to control some parameter(s) of the user's terminal or computer. The CCITT has provided for this by allowing a method of establishing an alternate path (kind of an in-band method of out-of-band signalling). This mechanism involves the use of the Qualifier, or Q-bit. The Q-bit is frequently used to provide the capability to a host to control a user's PAD parameters (such as to turn off echo when entering passwords). As far as I know, there is no easy form to do this in the datagram network, unless options are defined to do this.

Local Subnetwork Use

One of the clear advantages of the virtual connection networks is that it does follow the ISO reference model as far as subnetworks vs networks. The datagram network is good for what it is intended, an INTERNET protocol. Even the name implies that it hooks up networks and subnetworks to each other. IT IS NOT MEANT TO BE A SUBNETWORK PROTOCOL. What are we supposed to use within local subnetworks in the datagram network design??? TCP/IP works to interconnect subentworks, not act as the subnetwork protocol itself. Are we supposed to use just **link layer protocols** when communicating locally. **THIS IS** TOTTALY WRONG! I cannot emphasize this enough. **TCP/IP** on a subnetwork level makes absolutely no sense. It takes up too much overhead, processing speed, channel overhead, and memory requirements. Much grumbling was heard at first about the overhead of the address field of level 2 AX.25. Imagine if every packet must have an additional 40+ bytes of overhead to accomplish the same task. Some form of subnetwork protocol should be implemented, but **TCP/IP** is not it. Link connections such as what we use today also are a mistake.

A layered approach such as the virtual connection network design makes more sense. For the local subnetwork connections X.25 seems to fit real nice. It is a small robust protocol whose major defects don't affect performance at a local level. Since it is connection oriented similar to the presently implemented level 2 AX.25 protocol, plenty of the work has already been done.

The internetwork protocol of a virtual connection network would most likely be based on X.75, which is a modified version of X.25. Some additional work would be needed to make a complete network spec, but this would be fairly simple to **accomplish**. Since X.75 is also virtual **connection**, and it is a version of X.25, the two can be mapped together quite nicely.

The Transport Layer (if even required) is based on the CCITTX.224 standard (see another paper in these proceedings). X.224 is a **multi-class** protocol, and even the most basic class (**class 0**) handles the major hole in X.25/X.75 network operation (that of re-establishing a connection after a switch failure). A more advanced class also provides for a checksum to eliminate the possibility of a switch with a memory malfunction corrupting an otherwise accurately transferred packet.

Each of these protocols loads the major overhead burden into the connection establishment **process**, and then operates on a very small header **budget. One more point**, either the X.25 or the **X.75 protocols would be used** not both. This is to say that if a packet is originated in an X.25 subnetwork and then transferred across the amateur network using X.75, both headers are not required, just the one being used at that particular network connection.

Flow Controls

Flow control throughout the network is handled differently by the two network designs. The **datagram** network normally does not provide any flow control at the Network Layer. Instead it relies on the Transport Layer for end-to-end flow control, and the Link Layer for everything else. Unfortunalely, if the Link Layer is relied on, when the Link is flow controlled, not just the one network connection flow is stopped, but ALL LEVEL 2 data for ALL level 2 connections are stopped. Sometimes this is **alright**, but at other times this can be a big problem. There is no way around this problem.

In a virtual connection network, each individual network connection can be flow controlled independently of any other connection, independent of Level 2, independent of the Transport Layer. Some argue that this multiplicity of possible controlling devices adds unnecessary processing overhead and can lead to buffer problems stacking up and rippling through the network. I would point out that this most likely wouldn't happen, since there a finite number of packets **allowable in a network (in** either network design), due to Transport Layer sequence numbering constraints, in addition to **Level 2** sequence numbering constraints.

Circular File Philosophy

One of the comments I hear from time to time is that a **datagram** network is easier to implement, because of the capability of just tossing out a packet if it cannot be handled for any reason, and wait for a better time, or wait to see if the packet shows up again. I don't **feel** that the circular file is the place for my packets (some may disagree). I would prefer the situation that if a packet shows up, the network tries its best to get that packet **through**, and only if there is no other recourse (such as buffer limitations suddenly showing up) should the packet be thrown out or ignored. The **datagram approach** seems to rely on this "tossing the offending packet" instead of trying to correct the situation that caused the offending packet in the first place. I repeat, my packets belong in a better place than the trash heap.

Hardware/Software Considerations

An important consideration is what kind of hardware and software will be needed to run the two protocols. The biggest single requirement in both types of networks is going to be the requirement for lots and lots of RAM for buffers. The **datagram** type networks may need more buffers to be available at the end-points, while the need for more buffers in the virtual connection network may in the packet switches. It really depends on how the software is written as to how much buffering is required.

Another hardware/software consideration is that of processing requirements. This can be broken down into the individual devices that make up the network. The majority of the devices in the network **will** most likely be the packet switch. The **datagram** people claim that a **datagram** switch is easy to implement. Depending on the type of **routing used**, this may or may not be the case. If **some form of** dynamic routing is implemented, the **packet** switch suddenly becomes a much larger **device** requiring a lot more processor power to figure out the route the packet should take to reach its destination. Dynamic routing of some sort will probably be implemented in the **datagram** type network, since most of the advantages of the **datagram** network can only be taken advantage of in a dynamic routing enviroment.

A similar form of trade-off can be made in the packet switches of a virtual connection network, in a slightly different form. The first form is similar to the **datagram** approach. Full virtual connections are not maintained between every packet switch, but rather cross-connection tables are maintained at each switch **(similar to** the patch panel of an old phone exchange). **This** would allow very simple software to be implemented at the switches at first. The trade-off is that flow control can **only** be implemented at the Transport Layer or Link Layer (like the **datagram** network). If each packet switch implements a full X.75 network connection to each neighbor **switch**, processing overhead is increased, but the **overall** network becomes inherently more reliable.

The other device that must be considered **is that of** the network end-points. Here there is no question. Because of the need for a sophisticated Transport Layer protocol over a **datagram** Network Layer, the datagram network **will** require a substantially larger device with much more processing overhead. Distributed processing (one micro for each layer) may be an absolute requirement for datagrams, while an option **for virtual connections.**

An Ounce of Prevention...

Most amateur network users **will** always require that the network transfer data RELIABLY. The two forms of network designs place this responsibility in different places within the **network. The datagram** loads **ALL this** responsibility at the end-points of the network in the Transport **Layer. The datagram** Network Layer takes no responsibility whatsoever for maintaining **data integrity.**

The virtual connection network places this responsibility in small portions throughout the entire network, with the last margain of safety at the end-points in the Transport Layer. This distributed-responsibility scheme adds overhead throughout the network, but allows problems to be corrected along the way, rather than having everything look fine until it reaches the end-point, and only then finding out an error occured early in the network.

"What The Big Boys Use"

An issue that is sometimes raised is that of who is using what form of network. The research community seems to have fully adopted the TCP/IP datagram network concept, as provided by ARPANET. This is fully understandable, since they can quite often easily obtain the processing power necessary to implement TCP/IP. Also, since most of the research centers these days interract with the defense department who owns the ARPA network, there is some political pressure to go that route.

In the real world, the bottom line is the buck. The networks that are there not for research, but rather to provide the service of a data network (such as GTE Telenet) must look at how to provide a data network in the most cost-effective form, otherwise the competition will take their customers. It is interesting to note that the commercial networks use virtual connection protocols for their operation. In fact, Telenet was originally a datagram type network, but spent several million dollars to convert to a virtual connection network because they found out that the datagram network just wasn't cost effective. Some datagram people comment that the commercial data networks use virtual connection protocols because this shifts political network boundries out of the hands of the user and into the hands of the network. This seems to be based on articles in some of the computer journals around 1976. A lot has happened since then, including Telenet switching from datagrams to virtual connections. It is interesting to note just how many assumptions were made back then that are totally wrong today. Once again, the commercial networks use one yardstick for measuring their network, the biggest bang for the buck. No politics, because there is no room for politics. If they relied on political considerations, one of their competitors might not, and there goes the customers. It seems that the only people that can use political games are those that don't necessarily look at the bottom line, but can instead justify some additional costs for the sake of research. Does someone come to mind?

Conclusion

The major question I have for those implementing TCP/IP is what they are going to implement for the subnetwork (or intranet, or local network, or metropolitan network)? What are we supposed to use when packeting on a local basis to other hams in our area? Since a lot of our communications will always be within a metropolitan area, this issue MUST be addressed. Are we all supposed to support TCP/IP or UDP/IP? That won't work. You just can't shoe-horn all that on a TAPR board. Are we supposed to just continue to use Link Layer procedures when packeting locally? That isn't the right answer either. I believe that an AX.25 Level 3 machine could be shoe-horned into a TAPR board if one really tried.

As it appears from the above, I am going to continue the development of virtual connection network protocols. I do believe there will be a use for both network designs, and the best way to chose the correct one for the majority of the amateur network is to have both operate in a head-to-head competition. I do feel strongly that there is going to be a local subnetwork (intranetwork) protocol developed for local metropolitan users. This protocol does not have to be the same as the internetworking protocol used. In fact, I think there will most likely be some gateway operation to interconnect virtual connection networks with datagram networks. One point about this, I have heard some amateurs argue that if a part of a network is datagram then ALL of the network MUST be datagram (or vice versa>. This is not true!! All that must be done is that the gateway between the two types of networks must perform protocol conversions at both levels three and four. Since the two levels are so intertwined (especially with datagrams) this task must be accomplished. If it is done correctly, it should appear as if nothing out of the ordinary is happening.

My last comment is that given a piece of information that can be transferred using either method, which would you prefer and trust, the post office or the telephone system?

CCITT X.224 TRANSPORT LAYER PROTOCOL BASIC DESCRIPTION

Terry Fox, WB4JFI
President, AMRAD
1819 Anderson Rd.
Falls Church, VA 22043

Overview

In order to assure absolute data integrity through the amateur network, some form of transport layer protocol should be employed between the entry and exit points of the network. In datagram service, this transport comes in two basic forms, the Transmission Control Procedure (the TCP in TCP/IP) and the User Datagram Protocol, or UDP. The UDP is a very small transport protocol, and as such does not provide absolute data integrity under all conditions. The TCP is a much more robust protocol, and as such is capable of assuring absolute data integrity through the network, with a much higher overhead.

The Virtual Circuit amateur network concept using the CCITT standards (X.25/X.75) has generally relied on the use of the delivery confirmation, or D-bit procedures to maintain data integrity. Not only is this potentially a violation of the IS0 seven-layer model, but is also inadequate. If a virtual circuit connection is lost due to an intermediate packet switch malfunction, the D-bit procedures alone may not provide an accurate indication of what data was lost due to the malfunction. In addition, use of the D-bit provides no mechanism of detecting errors within the packet switches (such as memory errors) that might corrupt otherwise good data flowing through the amateur network.

This paper proposes the use of another CCITT X series protocol to correct for these potential deficiencies. This is a new recommendation, called X.224, and it describes a multi-class Transport Layer protocol that can be used on top of the X.25/X.75 Level 3 protocols. I will not give a detailed protocol specification here, but rather describe the different classes of the protocol and some of how they function.

Transport Layer Responsibilities

The basic function of the Transport Layer is to make sure that data traveling from the source end-point of a network to the destination end-point of the network does so in the proper order and without data corruption (if necessary). The Transport Layer relies on the Network Layer for providing a method of getting the data through the network from the source end-point to the destination end-point. Once the Transport Layer entity is sure that the data leaving it matches the data entering its peer Transport Layer entity, it will pass the data up to the next layer for further processing if required. The Transport Layer should have at its disposal some method of detecting errors, informing its Transport peer of these errors if necessary, and possibly correcting some forms of errors commonly encountered.

Errors Encountered By The Transport Layer

Before the X.224 Transport Layer recommendation is discussed, it might be helpful to describe some of the errors and situations a Transport Layer might have to deal with. Among these situations are:

A. Data Loss.
B. Data Corruption.
C. Data Duplication.
D. Data Misordering.
E. Data Misdelivery.
F. Network flow-controlled.
G. Upper layer flow-controlled.
H. Network Concatenation and Separation.
I. Segmenting and Reassembly.
J. Splitting and Recombining.
K. Multiplexing and Demultiplexing.

Data Loss

Loss of data can occur for a number of reasons in a network. If an intermediate packet switch goes down while holding data it acknowledged receiving but before sending to the next switch is one example. Total network failure is example of absolute data loss. These type of failures should be detected and corrected. This may mean a request for the missing data should be sent, or possibly tearing down the malfunctioning connection through the network and informing the user of the network failure.

Data Corruption

Data corruption is when data passes through the network from the source end-point to the destination end-point, but something happens to it along the way to create errors in it. While most of the time data passed through the amateur network will be passed from point-to-point using a reliable Link Layer protocol (such as AX.25), if an intermediate packet switch has a memory malfunction, data passed through that switch will be corrupted. Since the network corrupted the data, there should be some method of detecting and correcting this situation, if necessary.

Data Duplication

It is possible to have the same data delivered to the Transport Layer more than once. This happens most frequently when retransmissions due to loss of acknowledgements occur. In datagram networks can happen a lot when some type of flooding mechanism is used to correct network deficiencies. There should be some sort of detection of duplicated data, which results in the duplications being thrown out.

Data Misordering

With some network desi ns, it is possible for a group of data send be 4ore another group to arrive at the destination AFTER the second send group. This is common with datagram networks employing some sort of dynamic network routing scheme. Not only should this be detected, but some method of data re-ordering must be used to regain the original data order.

Data Misdelivery

Whenever the Transport Layer receives data from the Network Layer that it does not understand what to do with for reasons other than specified, it should have some method for acting upon the error. This can be as simple as ignoring the bad data, or as drastic as tearing down the Transport connection and notifying the user that the network has failed. This is kind of a catch-all for the "I'm so confused" feeling.

Network Flow-Controlled

While not an error, if not properly handled network flow control could cause the Transport Layer to become congested when the upper layer(s) keeps sending data that the Transport Layer cannot send over the network. Some method of flow-controlling the upper layer(s) should be employed to prevent data loss due to this potential situation.

Upper Layer Flow Controlled

This is also not an error, but rather a situation that could cause errors. If the upper layer(s) cannot accent more data from the network-through the Transport Layer, that upper layer(s) may try to stop further data from reaching it from the Transport Layer. If the Transport Layer is not capable of reacting properly, data may become lost between the layers.

Network Concatenation and Separation

Some networks allow the Network Layer protocol to concatenate more than one Transport Layer data unit into a single Network Layer data unit at the source end-point, and then separate the Transport Layer data units at the destination end-point. This may cause some problems at the Network/Transport Layer interface, which must be handled by the Transport Layer.

Segmentation and Reassembly

If the Transport Layer receives data from the upper layer(s) in a group or size that it cannot handle in a single Transport Layer data unit, it should break the data into multiple smaller Transport data units for sending through the network. The destination Transport Layer end-point should be capable of reassembling these samller data units into their original form before passing the data to the upper layer protocol(s). Some method of indicating this segmenting must be employed, along with some numbering scheme to make sure the data is reassembled properly at the destination end-point.

Splitting and Recombining

Occasionally Transport Layer protocols may use more than one network connection to support the same Transport Layer connection. If this happens, some of the above mentioned data errors may occur, especially data misordering. Special procedures may be required to support this function.

Multiplexing and Demultiplexing

This is when a single Network Layer connection is shared multiple Transport Layer connections. This will happen frequently, since it will result in reduced Network Layer overhead. Some method of assuring proper demultiplexing should be employed, otherwise the wrong user may get someone elses data.

There are more potential errors that can creep into the network system. The above highlights(?) some of the more common problem areas.

CCITT X.224 Recommendation

The CCITT has written Recommendation X.224 to serve as a Transport Layer Specification for Open Systems Interconnection networks. It is designed to be flexible enough to be used with a variety of Network Layers operating underneath it, while requiring a minimum amount of overhead.

Like most Transport Layer protocols, X.224 is designed to establish a "virtual connection" between two Transport Layer peers. In order not to ruffle the feathers of the datagramies, I will use the term "logical connection" in place of "virtual connection" throughout the rest of this paper.

One of the peers in this logical connection is called the source end-point, and the other is called the destination end-point. This is not meant to imply that data flows only in one direction, but rather that when an upper layer protocol sends a Transport Layer some data, as far as that data is concerned, it is at the source end-point of the Transport Layer connection. The source Transport Layer end-point then adds any overhead to the data that it may require for proper Transport Layer operation, and sends the resulting data to the Network Layer below it for transferring the data across the network to the destination Transport end-point.

The destination Transport Layer end-point then receives the data from its Network Layer, strips off and acts upon the overhead added by the source Transport end-point? and if everything is fine, passes the resulting data to its upper layer(s) for further processing.

As long as it is required to transfer data between network users, and no major errors occur, the logical connection between the two Transport Layer peers is maintained. Some errors can be recovered from using X.224 procedures, while others require tearing down, and re-establishing the logical connection, if it is still wanted after the error.

In order to provide one recommendation that is robust enough to handle different qualities of service provided by different networks, X.224 uses five different classes of protocols. This has the advantage of not requiring unnecessary overhead when using network protocols that provide a high degree of reliability, while allowing the overhead to grow to support network protocols that less quality of service.

CCITT X.224 Classes

The five classes of X.224 are:

Class 0. Simple Class.
Class 1. Basic Error Recovery Class.
Class 2. Multiplexing Class.
Class 3. Error Recovery and Multiplexing.
Class 4. Error Detection and Recovery Class.

The class of Transport Layer protocol used is inversely proportional to the quality of service provided by the Network Layer. If the network and the Network Layer protocol provides a high quality of service, then a simple Transport Layer protocol can be used. When the network service quality is poorer., a more sophisticated Transport Layer protocol might be required. There are three classifications of network quality defined in X.224:

Type A. Network operation with acceptable residual error rate (errors not signalled by disconnect or reset) and an acceptable rate of signalled errors.

Type B. Network operation with acceptable residual error rate, but unacceptable rate of signalled errors.

Type C. Network operation with unacceptable residual error rate.

Class 0.

Class 0 provides the simplest type of Transport Layer connection. It is designed to be used with type A network connections. It provides the functions necessary for logical connection establishment, data transfer with segmenting, and error reporting. Flow control relies on network provided flow control, and disconnection based on network service disconnection.

Class 1.

Class 1 provides a basic transport connection with minimal overhead, and is usually used with type B networks. The main purpose of class 1 is to recover from a network disconnect or reset. This is the basic class recommended for use in the amateur network over AX.25/AX.75 Network Layer protocols when absolute data integrity is not required.

Class 1 provides transport connections with flow control based on network provided flow control, error recovery, expedited data transfer, disconnection, along with the ability to provide consecutive transport connections on a network connection. In addition to providing Class 0 functions, Class 1 also provides the ability to recover from Network failures without affecting the network user. This is the main advantage of Class 1, and corrects for a potential problem in AX.25/AX.75 network designs.

Class 2.

Class 2 is designed to be used with type A networks. It provides a way to multiplex several transport connections onto a single network connection. Class 2 also allows the use of transport flow controls to help avoid congestion at Transport Layer connections end-points. No error detection or recovery is provided by class 2.

If the network connection resets or disconnects, the transport connection is terminated, and the user is informed.

Class 3.

Class 3 provides the characteristics of class 2 with explicit flow control, plus the ability to recover from network disconnects or or resets from class 1. This is usually used with type B networks.

Class 4.

Class 4 is the most sophisticated Transport Layer protocol, and is designed for use with type C networks. In addition to providing Class 3 services, it also detects and recovers from errors which occur as a result of a low grade of service from the network. The kinds of errors detected include:

A. Data Loss.
B. Out-of-Sequence Data Delivery.
C. Data Duplication.
D. Data corruption.

Detection and recovery from errors is enhanced by the extended use of data sequence numbering, timeout procedures, and a simple checksum system.

The class of protocol used can be negotiated at connection request time. If the preferred class is not available, an alternate may be selected, if appropriate.

X.224 Headers

Figure 1 shows the basic structure for the Transport Layer headers used in X.224. These headers are an integral number of octets long.

```
octet
 1    2  3  4  n   n+l           p  p+1
-----------------------------------------------
! LI ! Fixed Part ! Variable Part ! Data Field !
-----------------------------------------------
```

Figure 1. X.224 Transport Layer Header

LI is the Length Indicator field. It indicates how long the Transport header is in binary (less the LI itself). The maximum value is 254 octets (1111 1110).

The fixed part contains frequently occuring parameters, including the coding of the Transport Protocol Data Unit (TPDU) type. The length of the fixed part is determined by the type of data unit, protocol class, and format in use (normal or extended). The coding of the TPDU types are shown in Figure 2.

	TPDU Name	0	1	2	3	4	coding
CR	connect request	x	x	x	x	x	1110xxxx
CC	connect confirmation	x	x	x	x	x	1101xxxx
DR	disconnect request	x	x	x	x	x	10000000
DC	disconnect confirm.		x	x	x	x	11000000
DT	data	x	x	x	x	x	11110000
ED	expedited data		x	F	x	x	00010000
AK	data acknowledgement		C	F	x	x	0110zzzz
EA	expedited data ack.		x	F	x	x	00100000
RJ	reject		x		x		0101zzzz
ER	TPDU error	x	x	x	x	x	01110000
PI	Transport Protocol ID						00000001
	already in use by other protocols						00000000
							00110000
	not allowed in CCITT X.224						1001xxxx
							1010xxxx

Figure 2. TPDU Coding and Class Usage

Where:

xxxx indicates initial credit allocation in classes 2, 3, and 4. 0000 in classes 0, and 1.

zzzz indicates initial credit allocation in classes 2, 3, and 4. 1111 in class 1.

F Not available when non explicit flow control option is selected.

C Not available when receipt confirmation option is selected.

The variable part is used to indicated less frequently used parameters, if any are needed. The first octet of each parameter holds the parameter code, the second octet contains the parameter value length indicator, and the rest holds the parameter value itself. An example of use of the variable part is the use of checksums for the class 4 protocol.

The data field contains transparent user data, if any. Size restrictions depend on each TPDU type.

TPDU Header Structures

Figure 3 gives an outline of the structures of the headers used. It is not meant to be a complete description, just a brief indication of TPDU header size.

As mentioned above, the LI field is used to indicate how long (less the LI field itself) the Transport Layer header is.

The second byte contains the TPDU type, and optionally a credit field.

The DST-REF field is used to contain the address of the destination Transport end-point, and the SRC-REF holds the address of the source Transport end-point.

Connection requests and Connection confirmation headers contain a single-octet field that hold the preferred class the requesting Transport device wants to use, along with two option bits (use/non-use of explicit flow control and normal/extended formats in classes 2, 3, or 4.

Additional parameters may be requested in the variable part field, and are selected from among the following:

A. Transport Data unit size.
B. Version Number of Transport protocol.
C. Security parameters.
D. Checksum operation (class 4).
E. Alternative protocol class(es).
F. Acknowledge time adjustment.
G. Throughput adjustment.
H. Residual error rate.
I. Priority of data.
J. Transit delay.
K. Reassignment/resynchronization time.
L. Expidated data, additional options.

If a disconnect TPDU is sent, it should include a reason field which indicates why the disconnect was requested.

Data TPDUs contain a field that contains a bit (called EOT) which indicates when a TPDU is the end of a sequence of TPDUs. This field also contains the TPDU-NR, the TPDU send-sequence number. The TPDU-NR is set to zero in class 0, and may be any value in class 2 without explicit flow control.

Acknowledgement TPDUs contain a field called YR-TU-NR. This field contains the sequence number of the next expected data TPDU. It is used to indicate the last correctly received data TPDU to the source Transport end-point.

The last type of field used is the reject cause field in error TPDUs. As implied, it informs the source Transport end-point that the destination Transport end-point has rejected a TPDU, and why.

```
OCTET                  Connection Request Header
! 1 !    2    !    3    !    4    !    5    !    6    !    7    !    8    P! P+1          !
! LI ! CR ! CDT !      DST-REF      !      SRC-REF      ! CLASS,  ! VARIABLE ! USER DATA  !
!    !0110!     ! 00000000 00000000 !                   ! OPTIONS !   PART   !            !

OCTET                  Connection Confirm Header
! 1 !    2    !    3    !    4    !    5    !    6    !    7    !    8    P!  P+1         !
! LI ! CC ! CDT !      DST-REF      !      SRC-REF      ! CLASS   ! VARIABLE ! USER DATA  !
!    !1101!     !                   !                   ! OPTIONS !   PART   !            !

OCTET                  Disconnect Request Header
! 1 !    2    !    3    !    4    !    5    !    6    !    7    !    8    P! P+1          !
! LI !    DR     !      DST REF      !      SRC-REF      ! REASON  ! VARIABLE ! USER DATA  !
!    !1000 0000  !                   !                   !         !   PART   !            !

OCTET                  Disconnect Confirm Header
! 1 !    2    !    3    !    4    !    5    !    6    ! 7        P !
! LI !    DC     !        D          !                   !            !
!    !1100 0000  !                   !                   !   PART     !

OCTET
! 1 !    2         Class 0 and 1 Data Header        ...end !
! LI !    DT     !   TPDU-NR    ! 4  USER DATA            !
!    !1111 0000  !   and EOT    !                         !

OCTET               Normal Format Class 2, 3, and 4 Data Header
! 1 !    2       !  3  !  4  !     5      !  6         P! P+1      ...end !
! LI !    DT      !  DST-REF  !  TPDU-NR   !   VARIABLE  ! USER DATA       !
!    !1111 0000   !           !  and EOT   !     PART    !                 !

OCTET               Extended Format Class 2, 3, and 4 Data Header
! 1 !    2       !  3  !  4  !  5   6   7   8  ! 9         P! P+1      ...end !
! LI !    DT      !  DST-REF  !     TPDU-NR     !  VARIABLE  ! USER DATA       !
!    !1111 0000   !           !     and EOT     !    PART    !                 !

OCTET               Normal Format Class 1, 2, 3, and 4 Expedited Data Header
! 1 !    2       !  3  !  4  !     5      !  6         P! P+1      ...end !
! LI !    ED      !  DST-REF  ! ED-TPDU-NR !   VARIABLE  ! USER DATA       !
!    !0001 0000   !           !  and EOT   !     PART    !                 !

OCTET               Extended Format Class 2, 3, and 4 Expedited Data Header
! 1 !    2       !  3  !  4  !  5   6   7   8  ! 9         P! P+1      ...end !
! LI !    ED      !  DST-REF  !   ED-TPDU-NR    !  VARIABLE  ! USER DATA       !
!    !0001 0000   !           !     and EOT     !    PART    !                 !

OCTET     Normal Format Class 1, 2, 3, 4 Data Ack Header
! 1 !    2    !  3  !  4  !     5      !  6          P!
! LI ! AK ! CDT !  DST-REF  !  YR-TU-NR  !  VARIABLE    !
!    ! 0110 !    !           !            !    PART      !

OCTET     Extended Format Class 1, 2, 3, 4 Data Ack Header
! 1 !      2       !  3  !  4  !  5   6   7   8  !  9   10 ! 11         P !
! LI !     AK       !  DST-REF  !     YR-TU-NR    !   CDT    !  VARIABLE    !
!    !  0110 0000   !           !                 !          !    PART      !

OCTET     Normal Format Class 1, 2, 3, 4 Expedited Data Ack Header
! 1 !      2       !  3  !  4  !     5      !  6          P!
! LI !     EA       !  DST-REF  !  YR-TU-NR  !  VARIABLE    !
!    !  0010 0000   !           !            !    PART      !

OCTET     Extended Format Class 1, 2, 3, 4 Expedited Data Ack Header
! 1 !      2       !  3  !  4  !  5   6   7   8  !  9          P!
! LI !     EA       !  DST-REF  !     YR-TU-NR    !  VARIABLE    !
!    !  0010 0000   !           !                 !    PART      !

OCTET     Normal Format Class 1 and 3 Reject Header
! 1 !    2    !  3  !  4  !     5      !
! LI ! RJ ! CDT !  DST-REF  !  YR-TU-NR  !
!    ! 0101 !    !           !            !

OCTET                                 Class 3 Reject
! 1 !      2       Extended Format  5   6   7 Reject Header 10 !
! LI !     RJ       ! DST ! 4 REF !   YR - TU - NR   ! 9  CDT  !
!    !  0101 0000   !             !                  !         !

OCTET
! 1 !      2       !  3  !  4  ! Error  Header  . P!
! LI !     ER       !  DST REF  ! REJECT ! VARIABLE !
!    !  0111 0000   !           ! CAUSE  !   PART   !
```

Figure X.24 Transport Protocol Data Unit (TPDU) Header Formats

Procedures Available By Class

Figure 4 shows the various procedures of X.224, and in which classes they are available.

Procedure	Variant	Class 0	1	2	3	4
Assignment to network cnct		x	x	x	x	x
TPDU data transfer		x	x	x	x	x
Segmenting, reassembly		x	x	x	x	x
Concatenation, separation			x	x	x	x
Connect. establishment		x	x	x	x	x
Connection refusal		x	x	x	x	x
Normal Release	implicit	x				
	explicit		x	x	x	x
Error release		x		x		
Assoc. of TPDU with TCs		x	x	x	x	x
Data TPDU numbering	normal		x	m	m	m
	extended			o	o	o
Expedited data transfer	net normal		m	x	x	x
	net explic.		o			
Reassignment after fail.			x		x	x
Retention until acknowledgement of TPDUs	Conf. rcpt.		o			
	AK		m		x	x
Resynchronization			x		x	x
Multiplexing, de-muxing				x	x	x
Explicit Flow Control	with			m	x	x
	without	x	x	o		
Checksum	use of					x
	non-use of	x	x	x	x	o
Frozen references			x		x	x
Retransmission on timeout						x
Resequencing						x
Inactivity Control						x
Treatment protocol errors		x	x	x	x	x
Splitting and Recombining						x

X.224 Procedures

It is beyond the scope of this paper to describe the full operation procedures of the various X.224 classes. The above information is presented to show that there is an alternative Transport protocol to TCP that would work very effectively on top of an X.251X.75 type Network Layer protocol. I will continue to process the X.224 document into a form that will be presentable to amateurs, along with making sure that it is 100% amateur compatible.

The basic operational procedures of X.224 is designed to allow a logical connection to be established between a source Transport end-point and a destination Transport end-point using the connection request and connect confirm TPDUs, which may be passed along as part of the data field of an X.25/X.75 Ntework Layer fast-select connect request.

Once a connection has been established, data may flow in both directions, with optional flow controls, checksums, and sequence numbers. Optionally, expedited data can also be sent. Bad data can be rejected if necessary, and protocol errors can be detected and signalled.

When the connection is no longer needed, it may be terminated either explicitly, or by inference when the network connection servicing the Transport Layer is torn down.

Conclusion

I believe that X.224 is a viable Transport Layer protocol to use over an X.25/X.75 network. X.224 provides the small extra amount of protection over the X.25/X.75 network layer to insure absolute data integrity when necessary at a very low amount of overhead. Since X.224 is similar to AX.25 level 2 operation and X.25/X.75 network operation, an extensive software development campaign is not necessary to implement this protocol. Level 2 and level 3 protocol machines could be modified to provide much of the basic core of the X.224 protocol.

Interested users are encouraged to write the author for the latest in X.224 development, along with AX.25/AX.75 Network Layer development. It is also recommended to join AMRAD, as the AMRAD Newsletter contains fairly up-to-date information on packet radio development.

FADCA GATOR LINK 1 PACKET RADIO LINKING NETWORK

Howard Goldstein, N2WX
681 Cardinal St. SE
Palm Bay, Fl 32907

Ted Huf, K4NTA
1829 N.W. Pinetree Way
Stuart, Fl 33494

BACKGROUND

The GATOR LINK 1 concept was devised in the summer of 1984 by a group of members of the Florida Amateur Digital Communications Association (FADCA) as a method of linking packet radio digipeaters into a system that would provide wide area communications without the problems involved in single frequency digipeating. It was recognized that while AX.25 Level 2 provided a means of linking digipeaters, as packet radio activity grew, it would be more difficult to use this feature because of collisions.

RF EQUIPMENT

After a look at 23 cm as a possible band for linking, it was decided to use 1.25 meters because of the availability of equipment. The basic radio for the high speed side of GATOR 1 will be the Hamtronics FM-5 220 mHz. transceiver. The FSK modification done by Steve Goode will be incorporated to permit 9600 baud operation, GATOR 1 nodes will communicate point to point with up to three other GATOR 1 nodes. For this reason, directional antennas will be used, On the two meter side of a GATOR 1 node will use the normal two meter FM transceiver,

FREQUENCIES

A frequency plan using 145.01, .03, .05, .07 and .09 will provide 150 mile co-channel protection of two meter digipeaters in Florida. This will help relieve the common problem of digipeater to digipeater interference, All of the high speed 1.25 meter operation will be on 221.4 mHz. The Florida Repeater Council has coordinated 145.01, 221.4, 221.72 and 221.78 mHz for 20 kHz bandwidth and 220.57 mHz. for 100 Khz bandwidth state wide packet radio operation, FADCA has asked the FRC for additional coordination of 145.03, .05, .07, and .09 mHz. for state wide use with FADCA to serve as the coordinating body for all packet radio frequencies in Florida, No action has been taken by the FRC on this request yet.

COMPUTER EQUIPMENT

The brains of a GATOR 1 node will be the Xerox 820 computer with a FAD board, One af us (Goldstein) designed the FAD board, which uses a Zilog Z-8530A serial communications controller, This two port device will use one port for the 1200 baud two meter link and the other for the 9600 baud on 1.25 meters. The FAD board which is available from TAPR was first described in the June 1984 issue of the Florida Amateur Digital Communications Association newsletter, the FADCA>BEACON.

The modified Xerox 820 serves as a two meter digipeater for extending the range within a local area network, as well a5 a link into the GATOR 1 network from two meters and a relay station along the GATOR 1 network- Modifications to the 820 include removal of the GP PIO, disk controller and video chips, The modification includes the use of 2764 eproms. A FADCA Bell 202 type modem designed by Jerry Ruimby, N4AJH will be used on the 1200 baud side. And because of our commitment to provide emergency communication if needed, every effort will be made to provide GATOR 1 nodes with back up power. Figure 1 is a block diagram of a typical GATOR 1 node,

PROTOCOL HISTORY

The following is derived from the description of the proposed GATOR LINK 1 protocol that was published in the June 1984 issue of the FADCA>BEACON, originally written by one of us (Goldstein),

One of us (Huf) designed a system to assign addresses to these switches that nut only looks like an AX.25 level 2 address field, but also is easy for the end user to deduce without hard-to-find materials, Therefore, each GATOR 1 switch is identified by the area code (ala Fell) it serves in, addition to the airport designator (ex. 813TPA, 404ATL). When a switch hears a 1200 baud frame who's next outstanding address sub field matches it's own, it assumes that it is meant for relay and attempts to route it to an adjacent switch that is virtually closer to the destination along the connection it maintains with up to 3 of these GATOR 1 switches,

HOW IT WORKS

The information field in frames received from a high speed connection are parsed for their ultimate destination and get sent either to a switch closer to the destination area, or are passed to the

1200 baud transmit buffer if the local address matches the destination address.

EXAMPLES

Two examples of how an end user might set up a connection using the GATOR 1 system: MIAMI FL TO JACKSONVILLE FL. the user specifies: CONNECT N4UF VIA 305MIA 904JAX. Assuming the 305MIA switch hears me, it will take every one of my frames, set the H (has been repeated) bit on 305MIA and insert it into a virtual circuit with an adjacent switch that is nearer the 904 area code. Note that the routing is implicit, i.e. the path that the frames traverse is implied by the destination of each frame, and not by the user.

ORLANDO TO WEST PALM BEACH: CONNECT WA4HXZ VIA K4AHO 305ORL 305WPB WA4ARE-1. This example assumes that the AHO digipeater is needed in Orlando to reach the 305ORL switch, The same condition exists between WA4HXZ and 305WPB in this example.

The software machine that is the GATOR 1 switch was designed to fit into the 1K or so bytes left in the 2716 EPROM pair, and to run concurrently with the regular 1200 baud (low speed) digipeater code on the Xerox 820 (see Fig. 1 for block diagram), The A channel of the 8530 is used for the high speed side and the B port for the existing low speed channel. In this fashion interrupt priority is given to data flowing on the A channel, as servicing here must be accomplished much faster because the data rate is higher,

GATOR 1 PROTOCOL

The high speed links, i.e. the connections between adjacent GATOR 1 switches, use a virtual connection protocol similar to AX.25 level 3, but simplified. Each GATOR 1 switch will strive to maintain a connection with its neighbors by polling the current connection state and, should the connection be down, sending GATOR 1 connect frames to the appropriate switch, Once the connection(s) have been established with adjacent switches, they may now exchange data.

Frames passed between the switches are from 3 to 4096 octets long (excluding flag and CRC bytes). The shortest frames contain the source and destination CID (city ID) octets and a single command or status octet. Longer frames contain these plus an information field, which may contain multiple sub fields, each containing a complete copy of the relayed AX.25 level 2 frame,

CONCLUSION

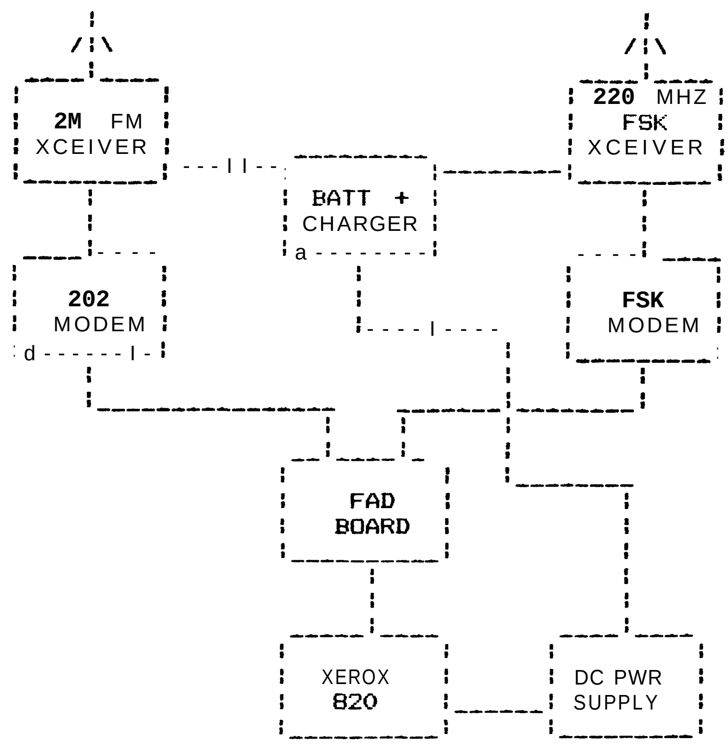
We realize that by proposing a linking facility that isn't "morally" consistent and doesn't even use the AX.25 Link-Layer it might seem we have rejected its use, This is not the case - in no way do we mean to discourage the proliferation of AX.25. Instead, the proposal simply reflects the current situation where there is no alternative for linking connections between different stations along anything but digipeat paths. The dilemma is in finding talented developers not of vaporware, but of software, and powerful and environmentally robust hardware that is inexpensive. If the latter situation was remedied, maybe some of the vaporware one hears about would congeal into something that works. Until then, the pragmatic aura of reality obscures all but that which exists, But here we find a very simple, talking alligator. It said something. ...Braaap????

FIGURE 2

```
        2 Meter 1200 baud                  220 High Speed

             ---                                 ---
             \ /                                 \ /
              !                                   !
            -----                               -----
            !    !-<-BCN                        !    !
          RBFR XBFR                           XBFR RBFR
            !    !-----<---HSROUTER--->----!    !
            !    !---<-------!    !--ll-<w--m-:
            !                !
          DIGIPEATER--->-----!
```

FIGURE 1



TYPICAL GATOR 1 LINK STATION

FIGURE 3

```
!FLAG!DEST CID!SRC CID!CMD/STAT!INFO!CRC!FLAG!
     where:     DEST CID,
                SRC CID are unique octets assigned
                 to a switch as its address
                INFO only where the CMD specifies an info frame,
                    formatted 1 i ke:
                  Offset       Comments
                  +0,+1         Length of following sub field (L)
                  +2• ..L+l     Data in sub field
                  {L+2,L+3       length of second sub field
                    ........3    info, ad nauseum up to 4096 octets


   CMD/STAT octet:
    !7!6!5!4!3!2!1!0
     u u ! ! ! ! s s------s s
         ! ! ! !          0 0 - Info frame (I)
         ! ! ! !          0 1 - Ack of Info (A)
         ! ! ! !          1 0 - Connect request (S)
         ! ! ! !          1 1 - Connect request ack'd (C)
         ! ! ! ----------- POLL/DONT POLLS when rcvd, returns an ACK if cnctd
         ! ! ------------- Connected/Connected*
         ! --------------- V (S)  }--Mod 2 seq-defined only for I frames
         ----------------- V(R)   }--Modulo 2 sequence number valid only
                                     for I and A frames
                           u - reserved, set to 1
```

TABLE Al

Link state	Frame received							
	I w/P	I wo/p	A w/p	A wo/p	s w/p	s wo/p	C w/p	c wo/p
0 setup					Cl	Cl 1	1	
1 xfer	Al	A1	Al		Cl	Cl		
2 wtng/ack	A2	A1	A2	A1	Cl	Cl		

TABLE A2

Link state	Condition		
	Tl expires	T3 expires	N2 exceeded
0 setup	SO		
1 xfer	A2*	A2*	
2 wtng/ack	A2*		SO

* INDICATES THE FRAME IS SENT WITH THE 'POLL' BIT SET

The format of actions in A1 and A2 are:

[T]s — where T is the type of frame to respond with [optional], and s is the state to enter after the indicated frame is sent

MODIFYING THE HAMTRONICS EM-5 FOR 9600 BPS PACKET OPERATION

STEVE GOODE, K9NG
140 W. WOOD APT. 314
PALATINE, ILL. 60067

Abstract

Within the last year considerable attention has been given to level three linking of local area packet groups across the country. Experiments using virtual and datagram circuits are under discussion. This paper presents an interface board and the necessary modifications to the Hamtronics FM-5 220 MHz transceiver allowing operation at 9600 bits per second (bps) to provide a high speed link between the local area networks.

Radio Selection

The purpose of this project was to provide a high speed radio modem as a testbed for the level three linking experiments. A minimum data rate of 9600 bps with minimum modifications to an existing radio were the main objectives. In order to send 9600 bps data with minimal overhead a radio with fast transmit to receive and receive to transmit turnaround time was desirable. This implied a crystal controlled transmitter with pin diode antenna switching. Since the ARRL band plan for 220 MHz contains allocations for high speed packet links a 220 MHz transceiver was chosen for the initial experiments. A radio that was still in production and available to all interested groups was also a prime consideration in selecting the Hamtronics FM-5 220 MHz transceiver. This unit contains a crystal controlled receiver and transmitter providing 7 watts of output power and pin diode switching of the antenna.

Receiver Modifications

The EM-5 is a EM transceiver so it contains a discriminator detector in the receiver. Frequency shift keying can be detected using a frequency discriminator so it was chosen as the modulation technique. The maximum data rate that can be sent thru the FM-5 receiver using FSK modulation was tested by frequency shift keying an RF generator with random data at different data rates and observing the received data on an oscilloscope. The pattern generated when the oscilloscope is triggered on the transmitted data clock is called an eye pattern. Figure 1 shows the eye pattern of the transmitted data on the top trace and the received eye pattern in the bottom trace for a 9600 bps data rate. One bit period is approximately two divisions in length. As can be seen in the picture, the received eye is open with minimal jitter at the bit crossings indicating that 9600 bps data can be sent thru the EM-5. Figure 2 shows the eye opening of 12 kbps data. The eye at 12 kbps is beginning to close with increased jitter at the bit crossings but the FM-5 could be used to send 12 kbps data. Figure 3 shows the eye pattern of 16 kbps data. This eye is closed with an unacceptable amount of jitter at the bit crossing. This data limit is due to the bandwidth of the IF filters in the FM-5 and could only be extended by increasing the bandwidth of the IF filters. Since minimum modifications to the radio were desired the maximum standard data rate was set at 9600 bps. The modification to the receiver is simply to connect a shielded wire to pin 9 of U1 (the discriminator output) which is run to post detection filters on the interface board.

The performance of the receiver at 9600 bps can now be tested by measuring the the bit error rate (BER) of the receiver. The BER of a data system is the probability of not receiving the transmitted bit correctly. This is normally expressed in percent or decimal form. For example, a system with a BER of 1 x 10-3 or 0.1% has the probability of receiving the transmitted bit incorrectly once in every 1000 bits. Bit error rate is measured by canparing the transmitted data bits with the received data bits and counting the number of errors. The BER for packet radio is dependent on the input signal strength to the receiver, so a graph of BER versus input power to the receiver is the normal measure of data system performance. Figure 4 shows the BER performance of the FM-5 receiver at 9600 bps. Figure 4 shows that at the 20 dB quieting level of the FM-5 receiver the BER was measured to be 4.5 x 10-3. We can compare this to data taken previously on the TAPR 1200 bps audio modem which was reported in QEX (1). Figure 5 is the BER performance of the TAPR 1200 bps modem. This figure shows that at the 20 dB quieting level of the Motorola Syntor radio the BER was measured to be 1.7 x 10-2. This shows that the 9600 bps radio has a system sensitivity that is approximately 1.5 dB better than the 1200 bps audio modem. The direct modulation of the RF carrier is a more efficient modulation technique than the audio subcarrier modulation of the 1200 bps modem providing a slight improvement in system sensitivity rather than a degradation which might be expected in going to a higher data rate. The QEX article shows that the AX.25 protocol begins to receive packets at the 10-3 BER and it is expected that any linking protocols will also begin reception at the 10-3 BER. We can then say that a EM-5 modified for 9600 bps operation will have a receiver sensitivity about 1 dB better than it's 20 dB quieting sensitivity.

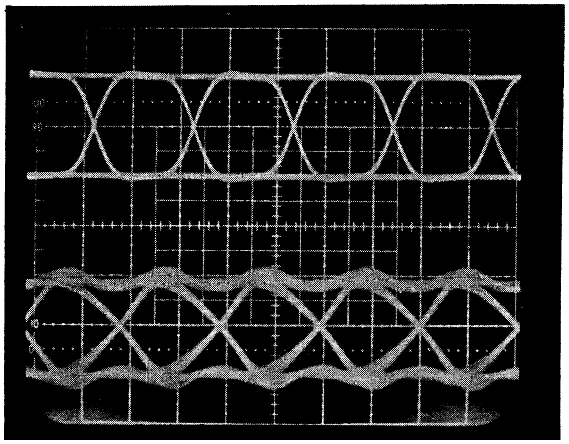


Figure 1
9600 bps Eye Patterns

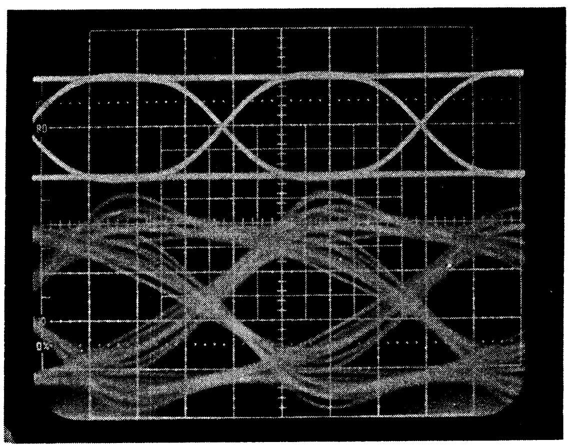


Figure 2
12 Kbps Eye Patterns

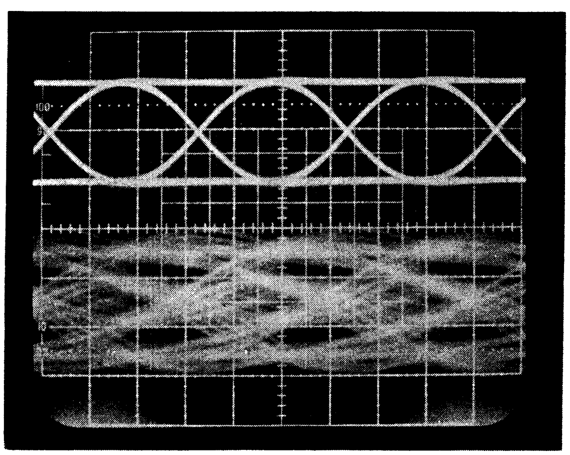


Figure 3
16 Kbps Eye Patterns

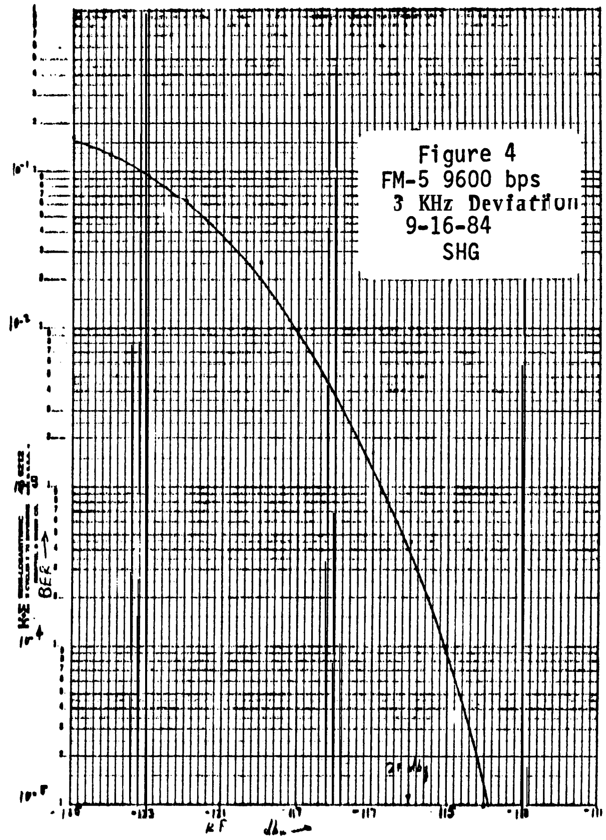


Figure 4
FM-5 9600 bps
3 KHz Deviation
9-16-84
SHG

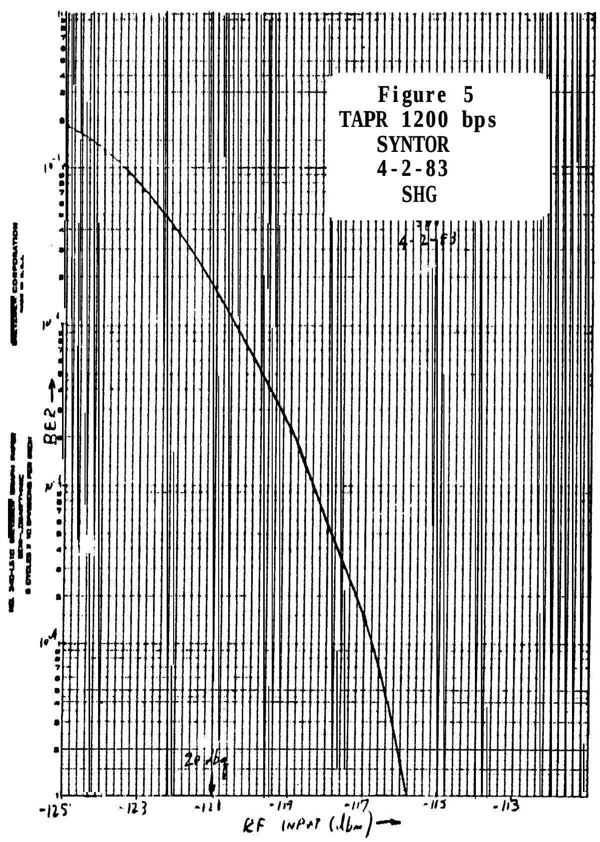


Figure 5
TAPR 1200 bps
SYNTOR
4-2-83
SHG

Transmitter Modifications

The Hamtronics EM-5 uses a phase modulator in the transmitter section. The transmitter was modified to provide direct frequency modulation by removing C25 and replacing it with the circuit shown in Figure 6. The varactor and shunt capacitor were mounted on the bottom of the board. The resistor and capacitor formed a flying connection to which a shielded lead was attached and run to the interface board, The FM linearity of this modulator was tested by measuring the frequency shift at the output of the transmitter with various DC voltages into the modulator. Figure 7 shows the FM linearity of this modulator. As can be seen, the modulator is linear over a 3 KHz range. A linear modulator was used in place of a frequency shift modulator for two reasons. First, since no modifications were made to the receiver IF bandwidth the resulting data system should fit into the present 40 KHz channel spacing 220 MHz band plan. This means that some method of controlling the transmitted spectrum must be provided in the transmitter. Figure 8 shows the resulting transmit spectrum when data is fed directly into a FSK modulator. At 40 KHz spacing there is still considerable energy which can interfere with the users on this adjacent channel. By filtering the data in a premodulation filter and then feeding it to a linear FM modulator the spectrum shown in Figure 9 is produced. This spectrum has greatly reduced energy in the adjacent channel. The second reason for not feeding data directly into a FM modulator without premodulation filtering is that the high frequency components of the data can excite crystal spurs resulting in the transmitter operating at many frequencies, some of which may be out of the Amateur band.

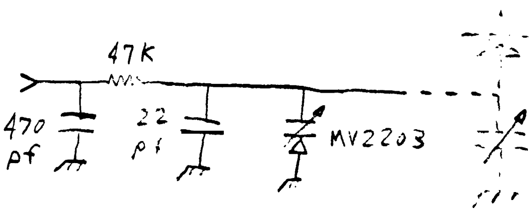


Figure 6
FM-5 Transmitter Oscillator Modification

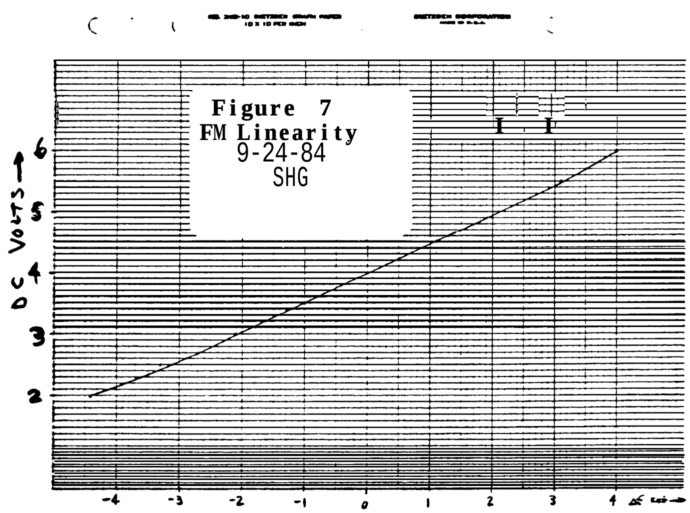


Figure 7
FM Linearity
9-24-84
SHG

Figure 8
9603 bps Transmit Spectrum
Unfiltered

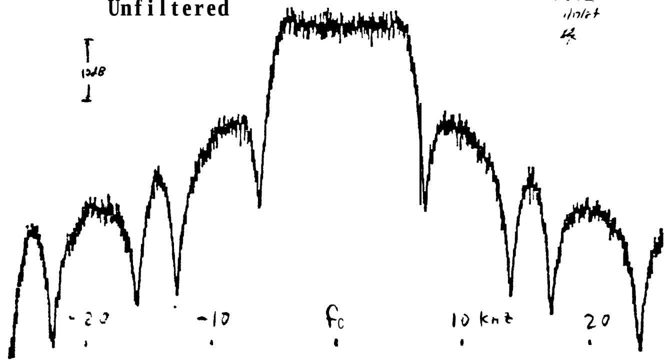
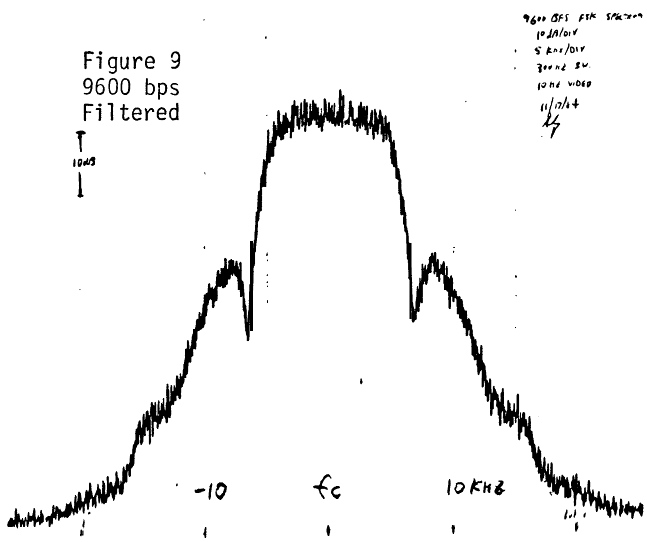


Figure 9
9600 bps
Filtered

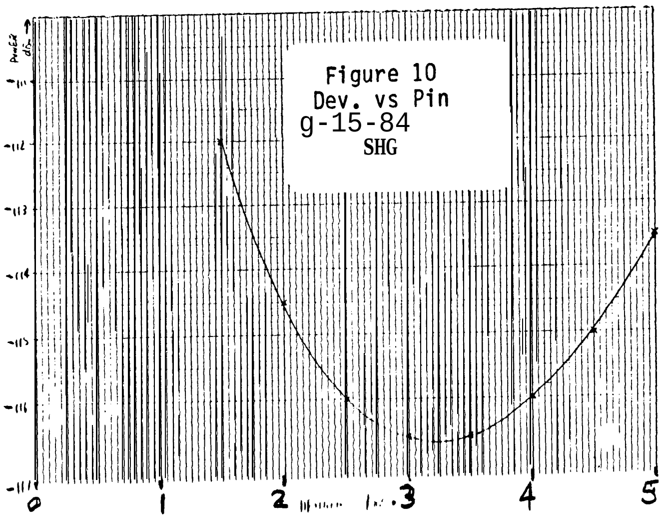


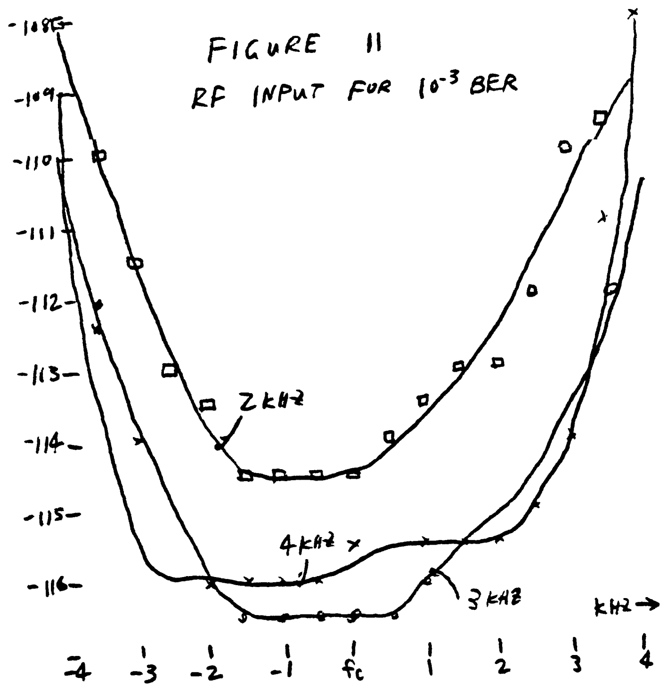
The optimum deviation for the 9600 bps data system was selected by measuring the RF power required for a 10-3 BER at various deviations. Figure 10 shows the RF power required for a 10-3 BER at 1.5 to 5 KHz deviation. Deviation was defined as the peak deviation when sending a 1-0 pattern. From this curve it is seen that the deviation for best sensitivity occurs at 3.25 KHz deviation. Since the data system should work with normal frequency offsets encountered with VHF radios due to temperature, aging and voltage effects, data was also taken measuring the effect of deviation on frequency offset. Figure 11 shows the RF power required for a 10-3 BER at 2, 3 and 4 KHz deviation with up to 4 KHz frequency offset. Taking into account the frequency offset performance of the receiver and the FM linearity of the transmitter the optimum deviation for the system was set at 3 KHz.

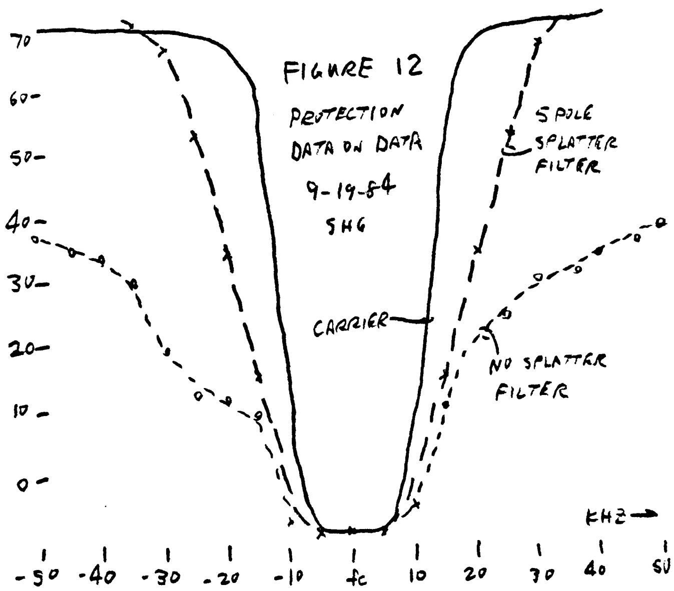
Now that we have defined the system deviation the protection given to adjacent channel users by the transmit splatter filter can be measured. Commercial manufacturers of voice radios measure adjacent channel protection in accordance to an Electronics Industries Association (EIA) specification (2) which requires a minimum of 70 dB protection to the adjacent channel. There are no specifications at the present time for data radios so we will have to make our own definition. The EIA voice spec essentially says to adjust the adjacent channel signal strength to sensitivity, then raise this level by 3 dB and degrade it back to sensitivity by the interfering adjacent channel transmitter. We can run adjacent channel tests by saying that the data system sensitivity is 10-3 BER. The premodulation filtered FSK signal was adjusted to give 10-3 BER and then this level was raised 3 dB. An interfering signal was then placed at various frequency offsets and the protection provided to this interfering signal was measured. Figure 12 shows the protection provided to a carrier, a premodulation filtered system (splatter filtered) and an unfiltered system (no splatter filter) at 9600 bps. As can be seen, 70 dB protection is provided to a carrier which should be similar to the EIA test. 70 dB protection is also provided to the described 9600 bps data system at 40 KHz channel spacing. Without splatter filtering only 34 dB protection is provided which can cause interference to operations on the adjacent channel.

An additional factor which must be considered in packet 'operation is how the transmitter is keyed up and down. Since this is done more often in packet operation than voice operation the frequent key up can produce a spectrum of its own that can be higher in splatter than the data modulation spectrum essentially negating the effect of the premodulation data filter. Once again there are no commercial specs for this frequent transmitter key up but a logical spec to set is that this key up spectrum should be below the data modulation spectrum. The interface board for the 9600 bps radio contains two DC key up switches which control this key up spectrum. The 9.1 volt line to the oscillator is keyed up 5 ms before the 12 volt line to the drivers and stays up for 5 ms after the drivers are turned off. This reduces the key up spectrum to an acceptable level. The modifications to the transmitter require removing C11, R3 and VR1. A separate 9.1 volt regulator with transmit switch is on the interface board and is brought into the radio in place of the removed parts.

A final modification to the transmitter requires removing R24. It was found that the charging of capacitors in the transmitter audio gave a spike at about 30 ms after key up to the phase modulator. This interfered with the data that was being transmitted at that time. Removing the audio from the phase modulator cured this problem.







Data Randomizer

Since VHF radios can have frequency offsets as large as the chosen system deviation for the 9600 bps data system, fixed decision levels for the received data cannot be set. This problem is remedied in this system by assuming a random data input and deriving the data decision level by averaging the received data. Unfortunately the AX.25 protocol does not provide random data so some method must be found to randomize the transmit data and then unrandomize the received data back to the original transmit data. This is provided on the interface board. A 17 stage shift register is configured to provide a self synchronizing data randomizer. This randanizer essentially divides the transmit data by a polynanial at the transmitter and then multiplies the receive data by the same polynomial at the receiver. The polynanial for this randanizer was selected to randomize AX.25 data sufficiently for transmission over the 9600 bps radios. Although this is not a standard maximum length polynomial it does sufficiently randomize AX.25 data with a minimum of extra parts. The randomizer has one drawback in that it requires 17 additional bits to be received correctly for the packet to be received. Since this is 17 bits in addition to the over 1000 bits in a packet this does not significantly change the BER required for the system to begin receiving.

Interface Board

Figure 13 is a schematic of the interface board required for the 9600 bps modifications to the Hamtronics FM-5. Ul is a quad op amp which provides the 5 pole 5 KHz Bessel filter used for the transmit premodulation filter and a 2 pole 5 KHz Butterworth filter used as the post detection receiver filter. The filters were designed to give 70 dB adjacent channel protection at 40 KHz in the transmitter and to provide optimum BER performance in the receiver. U2 and U10 provide the 12 v and 9.1 v key up controls. U5 and U6 form the data randanizer with U7 providing the necessary switching between transmit and receive. U4 and U3 form a clock recovery circuit necessary for proper data randomizer operation. A clock lock output is also provide by U4 and U3 which is used as the data carrier detect function. U8 provides a 1-0 pattern to the transmitter during receive to keep the oscillator on center frequency and ready to transmit. U2 provides a time out timer for the transmitter and the data slicer for the receiver. Channel frequency and deviation of the radio are adjusted by removing jumper 1 and shorting jumper two. C26 in the FM-5 is then adjusted for center frequency. Jumper 1 is then replaced and the deviation on the 1-0 pattern is set for 3 KHz.

Field Tests

An FM-5 transceiver was modified as described in this paper along with a Hamtronics T-51/R-220 exciter receiver pair. The interface board was mounted external to the FM-5 and all control leads were run thru a 9 pin D connector mounted on the back of the radio. The exciter/receiver pair used the same interface board as the EM-5 but required the building of a pin diode antenna switch. Field tests were then conducted between K9NG in Palatine, Ill. and W9TD in Hoffman Estates, Ill. which is approximately a five mile path. K9NG used the 7 watt FM-5 and W9TD used the 1 watt T-51/R-220 pair. Two TAPR kit boards were jumpered for twice clock operation and connected via the modem connector. 9600 bps packets were then sent between the two stations. Both stations used a 1/4 wave ground plane antenna. These field tests confirmed the bench tests of the 9600 bps radios. The Txd of the TAPR boards were set to 1 resulting in a 20 ms delay for transmitter key up. 128 byte packets were sent in about 150 ms. These packets just open squelch in a normal FM receiver and sound like noise bursts. 100 Kbyte files were transmitter in approximately one and a half minutes.

Modifying Other Radios

The interface board presented here can be used to modify other radios for 9600 bps packet operation. As was mentioned in the field test results, a Hamtronics T-51 and R-220 exciter/receiver pair were also modified with similar results to the FM-5. One area that may cause a problem with other radios is the transmitter key up. Since other radios may key up differently than the EM-5 or T-51 the method of controlling the key up used for these radios may not work for others. If at all possible, the key up of the intended radio for modification should be observed on a spectrum analyzer and verified to not create a large transmit spectrum when keyed up at an 8 Hz rate.

Comments on Higher Data Rates

The filters on the interface board can be directly scaled for higher data rates when radios with larger bandwidth IFs become available. It should be realized that any increase in data rate from the 12 kbps maximum allowable in the standard 15 KHz bandwidth of an EM receiver will degrade system sensitivity in a direct relationship. For example if a 96000 bps system was desired the IF bandwidth would have to increase by approximately 10 times and a 10 dB degradation in system sensitivity would be experienced. This means that link station separation would have to be decrease or transmitter power would have to be increased to make up for this degradation in system sensitivity. The apparent free lunch in system sensitivity that was obtained by going to direct FSK will not be available for higher data rates and the degradation in system sensitivity will be a tradeoff in designing link stations to use these higher data rates.

Conclusions

The necessary interface board and modifications to allow packet operation of the Hamtronics FM-5 transceiver at 9600 bps have been presented. Field tests have proven the design to be workable. Every attempt has been made to design this radio interface to be a good neighbor in the 220 MHz band. Adjacent channel protection ratios were measured for the system and provide a minimum of 70 dB protection to the adjacent channel users.

Since the EM-S provides a maximum of 7 watts of output power, linking groups may be interested in designing a power amplifier. This power amplifier would have to be designed to provide fast antenna switching and proper key up. Future projects could also include a radio designed specifically for 9600 bps packet operation or higher data rates.

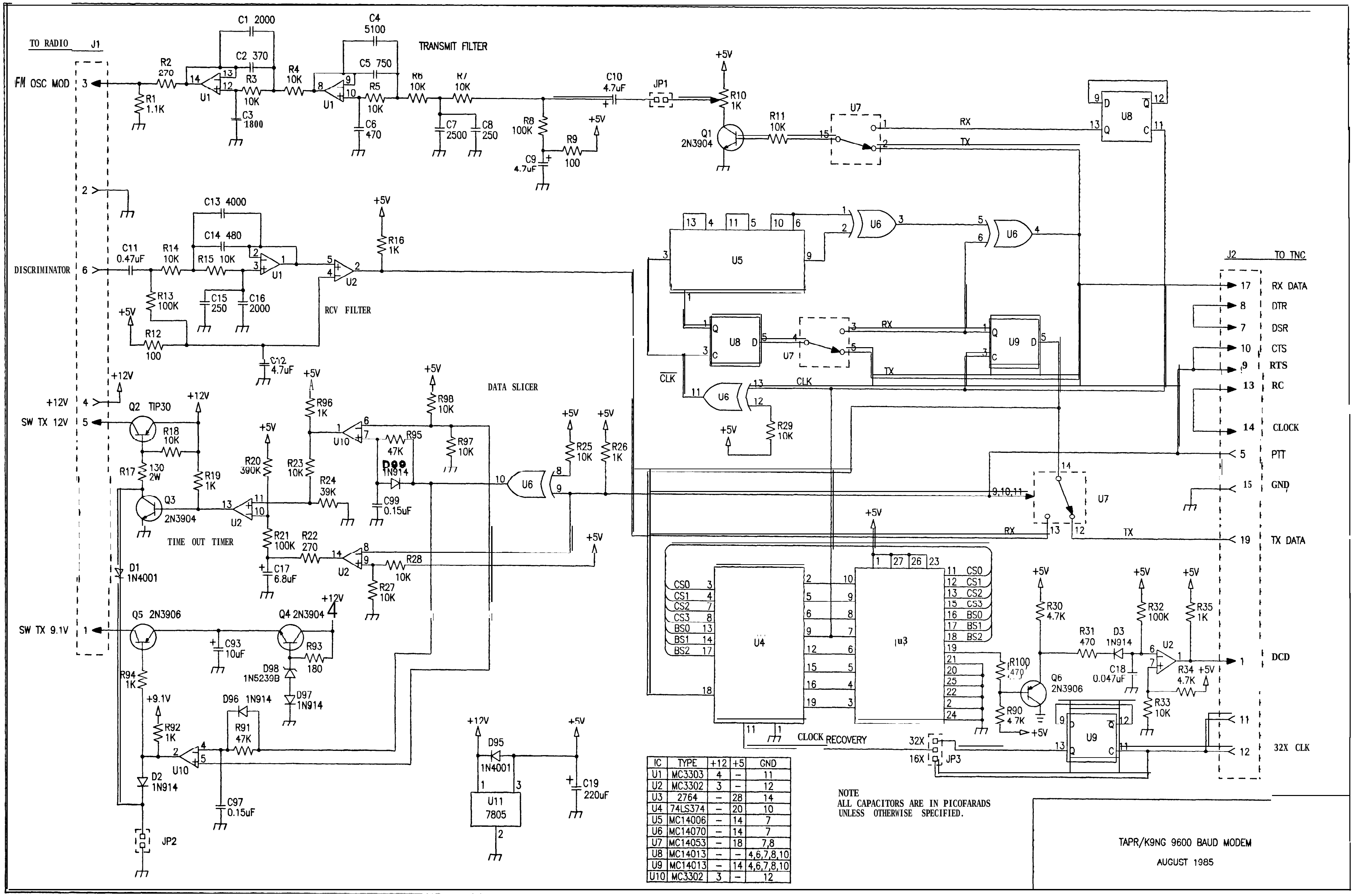
Acknowledgements

I would like to thank Paul Newland, AD71 for providing the state machine code used in U3 which implements clock recovery and data carrier detect. I would also like to thank Gary Kaatz, W9TD for performing the 9600 bps field tests. Finally a large thank you goes to Pete Eaton, WB9FLW for originally suggesting this project and to Lyle Johnson, WA7GXD and the rest of the TAPR gang for their encouragement throughout the project.

References

1) Goode Steve, K9NG, "The Bit Error Rate Performance of the TAPR TNC Modem", QEX 18, August 1983

2) Electronics Industries Association, "Minimum Standards for Land Mobile Communication FM or PM Receivers,25-947 MHz", RS-204C, January 1982



IC	TYPE	+12	+5	GND
U1	MC3303	4	–	11
U2	MC3302	3	–	12
U3	2764	–	28	14
U4	74LS374	–	20	10
U5	MC14006	–	14	7
U6	MC14070	–	14	7
U7	MC14053	–	18	7,8
U8	MC14013	–	–	4,6,7,8,10
U9	MC14013	–	14	4,6,7,8,10
U10	MC3302	3	–	12

NOTE
ALL CAPACITORS ARE IN PICOFARADS UNLESS OTHERWISE SPECIFIED.

TAPR/K9NG 9600 BAUD MODEM
AUGUST 1985

PACKET RADIO FOR DISTANCE TEACHING IN THE THIRD WORLD

Phil Gray, KA7TWQ
International Council for Computers
in Education
Home: Box 731
LaGrande, OR 97850

Abstract

How Packet Radio could greatly improve the interactive capabilities of Distance Teaching. Some views on how Packet Radio could improve the delivery and efficiency of Distance Teaching in Less-Developed Countries.

Definition

One problem every nation faces is delivering *some* sort of education to its citizens no matter where they are: those in remote or sparsely-populated areas; "those severely constrained by religious or other traditional social factors (e.g., women, or members of low caste in some countries)."[1]; the handicapped; refugees; prisoners; etc. One means of dealing with this situation is the use of Distance Teaching. Distance Teaching is defined by Joseph N. Pelton as a "multimedia educational process in which the teacher and learner may never meet face-to-face. . . . This form of education has emerged largely because society has not been able to satisfy needs through traditional educational structures."[2] Distance Teaching is not new--even to this country--but it seems to be gaining renewed prominence, especially with the advent and promise of satellites. (The reader is referred to OSCAR 10 experiments last year,[3] and the UoSAT-OSCAR 11 proof-of-concept demonstration in January this year.[4]) While Distance Teaching is of increasing importance in the Developed Countries, this report is aimed at the Less-Developed Countries.

Interactive Distance Teaching

For Distance Teaching to really reach its full power, it needs to approach as near as possible the face-to-face, interactive conditions of teacher-to-student. Packet Radio makes this goal the most affordable yet. Because of the following important features, I believe Packet Radio can bring interactive Distance Teaching to a point educators never even *hoped* for before:

1. Costs can be held down with the bare bones, minimum components as follows: a personal computer and display of some type, a Terminal Node Controller, a transceiver, and an antenna.

2. Brands of computers can be mixed because the use of ASCII code is the great equalizer between computers of different types.

3. Enough Digipeaters (Packet Radio stations as detailed in the first item, above) can make the system's range nearly limitless.

4. This medium is digital, almost 100% error-free communication. This opens up whole aspects of information transfer that were simply not feasible prior to this time. Today, if it can be digitized, it can be sent and, if it can be sent errorlessly, then data transfer suddenly becomes a far more valuable and cost-efficient exercise than before.

Price

Cost of a minimum Packet Radio system will vary according to hardware sophistication, features, and whether peripherals are added. I am currently conducting a survey of Packet Radio users, but it is too soon to publish any reliable information. However, the cheapest arrangement reported so far was $525 with some used components and the most expensive was $2,125. The average--again, from a very small sample--is $1,025.

It should be noted, too, that a savings can be made if the Terminal Node Controllers are purchased in kit form and assembled on site. This would be an excellent exercise for a science or electronics class. For nations that permit amateur radio activity, it might be possible to obtain volunteer time and equipment such that initially only the Terminal Node Controllers need be purchased. This would be an especially good approach if a government wished a trial of the Packet Radio concept before committing to it.

One of the nice surprises in store for an agency considering the costs of a Packet Radio installation is no new facilities need be constructed. A Packet Radio station requires no more space than a large closet--so any packet station could reside in the main ministry of the capital as well as the smallest classroom in the country.

Lastly, when calculating costs, it must be remembered that once the Distance Teaching chores are over for the day, the equipment should never sit idle. The hardware can be utilized in many additional, exciting, and valuable ways. For a classroom entering the electronic age, here are two perfect instruments to be considered as tools, as objects of study themselves, and as assistants in the classroom: the computer/display and the transceiver/antenna.

Applications: Interactive

Here are but some of the possible uses of Packet Radio for interactive Distance Teaching:

--Upgrading, improving, or enhancing the skills of the teacher/aide already on site. A master teacher in the central district could provide lessons for presentation by an instructor in some isolated area. This could inject new ideas *or* skills into the repertoire of a teacher/aide/tutor. The person in the remote part of the country would receive instant feedback regarding the correct use or presentation of the lesson prior to giving it to the class. This is the type of interaction that improves teaching efficiency: sort of a "remote in-service."

--Monitor students' academic progress with daily or hourly checks for rapid diagnosis and remediation. Again, the experienced teacher interacts from a distance to assist the new or unsure teacher on how best to proceed with a student's lessons.

--Personalized lessons based on students' performance (with individualized instruction on subjects as varied as reading or engine repair).

--Academic credit. Students could "take classes from schools all over the country . . . even earn high school, college, and graduate level credits through the system."[5]

--Timely contact with the teacher from the provincial or capital offices. This would facilitate important information reaching the teacher quicker than the postal system (which is slow enough in Developed Countries!). Bulletins, pertinent literature reviews, even disaster warnings could reach the school with ease and reliability.

--Sending up-to-the-minute information from the teacher back to the central office. This would provide the up-country instructor with a means to obtain advice, support (moral or pedagogical), or emergency assistance in a most expeditious way.

--Teacher-to-teacher contact and support. Communication with one's colleagues is important in any endeavor and especially when one is new or inexperienced to the job. With Packet Radio, an opportunity to seek the company of someone else "in the trenches" is readily available 24 hours a day. For this (and for digipeating), a Packet Radio station ideally should remain up-and-running every hour of the day where possible.

--Bulletin board service. With the availability of a BBS, many of the above types of communication would be enhanced.

Applications: The Computer

No matter what make or model it is, there are a myriad of things a personal computer can do for the teacher. As a tool (depending upon configuration, memory, and time available for use) it can inventory, keep records and grades, perform EMAIL and store and forward messages, store programs, control environmental conditions of the classroom/school, and word process. "Information retrieval may be used . . . directly by students as a quasi-library facility. . . . [And, not least,] students interact with the computer without intermediaries for the express purpose of increasing their knowledge or skills in some school subject."[6]

As an object of study, there are spreadsheets and word processors, programming languages, input/output, data retrieval or computer-to-computer communications, simulations, graphics, artificial intelligence, etc.

It can become an assistant to the teacher by taking small groups of students aside to work or play while the teacher does large-group instruction with the rest of the class. It can also free the teacher by providing tutoring, drill, and practice or review, not to mention administer tests of placement or comprehension.

Applications: The Radio System

Here--to a much lesser extent--we also have a tool (depending upon the electronic skill and confidence of the operator). It can locate other sites, page people, perform RTTY and Slow Scan Television, etc. As an object of study it is perfect for analog and digital electronics, antenna theory and construction, Orbiting Satellite Carrying Amateur Radio (OSCAR) and the necessary tracking (great for applied math), etc. As an assistant it too can separate a small group of students off to entertain or instruct.

It must be mentioned this new wonder of our lives--Packet Radio--need not, indeed *should* not, be limited to use only by educators in the school. Presentations could be done after hours that offer courses, lectures, or updated developments of interest to other members of the surrounding area such as farmers, merchants, hobby groups, investors, etc.

Other Considerations

Before embarking on this path for improving Distance Teaching, three factors must be considered: environmental, sources of power, and human resources.

Environmental: As great and wonderful as computers are, at this stage in their evolution, they can quickly be felled by

> heat, dust, humidity, vibration, and mechanical shock. . . . they are not often moved, and when they are, are moved with care. . . . Environmental constraints can be compensated for in two ways: the environment itself can be controlled by means of air conditioning, etc. or computers that are more resistant to environmental hazards can be purchased; such computers are available, having been developed for the Armed Forces for use in uncertain field conditions, but are quite expensive in comparison to the commercially available computers.[7]

On the other hand, lap board computers, such as the Radio Shack Model 100 which have all the

necessary components enclosed, could be a distinct possibility. The only drawback is a small display and limited memory for storage, but even the *memory* can be expanded with cassette tape and recorder.

Sources of Power: Many schools in remote areas of the world are without power. Those that are blessed with it often find it unreliable or fluctuating more than computers can tolerate. Alternatives have been studied, especially in regard to "appropriate technology.** Agencies such as the Peace Corps, Volunteers in Technical Assistance, and the Amateur Radio Relay League actively seek new and inexpensive designs and creations in the area of power supplies for electronic equipment. Various types of generators, battery configurations, and solar panel arrays exist now and the market grows at a rapid pace.

Human Resources: Whether it is the instructors themselves, aides, or dedicated operators, some sort of training must occur before a system such as this could be put in place. A way to forgo this expense of time and money would be to limit the placement of the hardware to the classroom or school which had a science teacher on site. Even so, I contend the training necessary does not have to be extensive or lengthy because, as hardware engineering continues to output more and more sophisticated equipment, the operator needs to know and do less and less. Add to that a good software program to achieve minimum levels of operator interaction and the prerequisite skill (if not anxiety) levels would further reduce.

Conclusions

Packet Radio most certainly should be considered as a delivery system for Distance eaching. Its technology could make Distance Teaching truly interactive in every sense. Its costs are certainly competitive with any present-day delivery system and its future use via satellites is ensured. Added to that, a Packet Radio station site has a computer and transceiver as tools and teaching aids when not in use with Distance Teaching. Most Distance Teaching projects are aimed at adults, but I must emphasize there's no reason primary-age students could not be well served by the equipment and concepts in this proposal.

References

[1]M. W. Neil (Ed.)., *Education of adults at a distance: A report of the open university's tenth anniversary international conference* (london: Kogan Page, Ltd., 1981), p. 65.

[2]J. N. Pelton, & R. T. Filep, Tele-education via satellite. In W. T. Blume & P. Schneller (Eds.), *Toward international tele-education* (Boulder, CO: Westview Press, p. 158), p. 158.

[3]QEX, *The ARRL Experimenter's Exchange* 26 (April 1984): 1.

[4]UoSAT-OSCAR 11 demos PACSAT concept, *Amateur Satellite Report* (January 28, 1985; No. 94/95): 1.

[5]R. Cordon, Welcome to the electronic university! *The Electronic Catalog* (Fall 1984): 3.

[6]J. Friend, *Classroom uses of the computer: retrospective view with implications* for *developing countries,* unpublished paper (December 1984): 2.

[7]Ibid., p. 23.

Packet Radio Development - 1985

by Lyle V. Johnson
President
Tucson Amateur Packet Radio
PO Box 22888
Tucson AZ 85734-2888

Abstract

A review of packet growth since the Third ARRL Networking Conference is followed by a discussion of anticipated expansion of packet activity during the next year.

A framework for orderly growth is presented, based on the above observations.

1984

Orwellian associations not withstanding, 1984 was a year which saw tremendous growth in Amateur packet radio in the United States as well as the rest of the world. This growth included technical advancement in addition to a vastly swelled user base.

On March 1st, UoSAT/OSCAR-11 blasted into orbit, carrying a PACSAT-like prototype Digital Communications Experiment (DCE). The DCE was publicly demonstrated at the Pacific Telecommunications Conference in Hawaii in January, 1985, via a store-and-forward technique. Communications were supported between England, California and Hawaii during this operational test.

In mid-summer, the 23rd Olympiad was hosted in Los Angeles. The Football games were held in Stanford, near San Francisco, and Amateur packet radio was used to carry hundreds of messages related to the Stanford events.

HF packet has been used on 40, 30 and 20 meters to provide a primitive linking capability between Arizona, Massachusetts Washington, D.C., and other areas.

Meteor scatter techniques were tested on 6-meters, resulting in reliable, if slow, data transfer between Iowa and Washington, D.C.

The WORLI bulletin board/message forwarding system has gained widespread acceptance in the packet community, furthering the "networks without networking" experimentation.

The sheer number of packet radio participants has increased between four-and tenfold, depending on whose figures you believe. TAPR alone placed in excess of 100 TNCs per month during calendar 2984, and the rate has not slackened as of this writing.

The ARRL Ad Hoc Digital Committee, working with in&rested packet groups, has been sponsoring a lively debate and results-oriented "contest" between the two major viewpoints for Networking -- Virtual Circuits and Datagrams, the former represented by the AX.25 Level Three approach and the latter by TCP/IP.

Early 1985

1985 has opened with a "bang" for packet radio. The two fronts of packet expansion, technical advancement and marketing, have been addressed, and rather dramatically.

Stave Goode, K9NG, of the Chicago Area Packet Radio Association, has developed a 9600 bps modem capable of working within a 20- to 40-kHz bandwidth using straightforward direct FSK techniques. Steve's work opens the door for widespread use of 9600 bps (and faster) packet data channels through the virtues of simplicity and economy. In conjunction with this effort, TAPR is dedicating significant resources to the development of an integrated modem/rf deck for 9600 bps packet operation on the 220-MHz Amateur band.

On the marketing front, a major manuafacturer of Amateur radio equipment, Heathkit, has entered the packet fray. Unveiled at the Miami Tropical Hamboree, and later at the TAPR Annual Meeting, both in February, 1985, Heath has produced a "TAPR-clone" TNC kit to sell for under $300.

Heath has indicated that a major share of their current revenues are generated by their computer product line, and packet is a logical way to link the computer and amateur radio markets. While this may seem obvious to most packeteers, Heath is the first manufacturer with a significant presence in both markets (and perhaps the only manufacturer in that category!) to commit resources to the packet marketplace.

There are indications that other manufacturers may be entering the Amateur packet radio arena; at the least, it seems reasonable to expect innovative, alternative packet hardware and software to become available during 1985.

What Does AU This Mean?

Packet information available to TAPR **suggests** that there is starting a considerable influx of newcomers to Amateur packet radio. Many of these people are non-technically oriented. Their interests range from traffic handling, through emergency communications, **to simple** curiosity.

As equipment becomes available that **is** easy to integrate in the average ham shack, and documentation is written to make packet operation easy to understand, the influx of less technical members of the Amateur community to packet radio will likely increase. In many areas of the country, local packet activity has seemed to reach a critical mass, with newcomers appearing on a weekly, sometimes daily, basis.

These people want to operate packet, not develop it. They aren't interested in a network that doesn't exist, or potential that is untapped. An organizational structure not unlike the present VHF and UHF repeater system may emerge, however, with a user community willing to assist in funding a communications system from which they will derive direct benefit.

We are reaching *a* point in time that will require packet radio to deliver on its promises.

Thus, we **forsee** *a* significant impact on Amateur packet activities from the operational standpoint.

On the technical front, we find that many areas, particularly those areas that make extensive use of the digipeating facilities offered by the AX.25 Level 2 **protocol,** are experiencing saturation. This results in long delays, multiple retrys and other assorted negative factors.

While 1200 bps is *a* significant advance over other widely used Amateur digital **signalling** rates, it is unreasonable to **expect** this sort of bandwidth to **accomodate** a large population, especially if confined to one or a few channels.

Further, **as** initial Network Level protocols are implemented and more packet stations *are* able to access the facilities offered on these networks, congestion is bound to increase.

Thus, **we** see a significant impact on Amateur packet operation from a techncial standpoint.

A Plan For Growth

We are familiar with much **of the** potential of ***packet.*** When multiple, **medium-** to high-speed links blanket the continent, we will be able to dump megabytes' worth of data with ease and confidence. Real-time video, **inter-**computer communications -- the sky's the limit.

But this is only potential. If we invoke this vision too often to the uninitiated, they will begin to believe we are speaking of the present, not **some** (hopefully not distant) future scenario. Disallusionment can only feed the fires of the scoffers and detractors.

The problem at hand is how to build to this kind of a system in **manageable** steps, each step taking us closer to our eventual goals, **yet** with the efforts having long-term **usefulness** as well **as** near-term **effectiveness.** And **of** course, the bill to develop and produce these evolutionary goals must be small enough to be absorbed by the packet community existent at the time the step is taken.

We do not presently have the manpower, technical experience nor money to put up a blanket-the-nation high-speed network. **We** don't even have protocols tested in the Amateur environment upon which to build such a system. A suggested course of action is:

1) Make a decision on Network Level protocol. Work together to implement this protocol, prefereably on standardized hardware that is both capable of supporting Network and Transport decisions and doing so on multiple channels running **at** 9.6 kbps to **56** kbps.

2) Design rf decks that are **easily** built and **adjusted,** low in cost, and capable of operating with existing packet radio controller equipment at 9.6 **kbps.**

3) Get multiple channels coordinated and operational between metropolitan areas that have packet communities to support them, Such channels should operate at **a** data rate of at least 9.6 kbps. Future operation at 56 kbps should be planned. Eventual operation at 256 kbps to 2 **Mbps** should be anticipated.

4) Establish HF gateways in **major** area8 of packet operation. 300 bps/200 Hz shift has become standard on **40/30/20** meters in the US. Petitioning of regulatory bodies to allow operation **at** 1200 bps should be done at the earliest practical date, and waivers to allow technican-class licensees digital traffic to be "linked" on HF frequencies **should be** requested.

5) As regional networking **occurs, *a* si**multaneous national effort should be undertaken to develop and fund sites that do not have the packet population to support a local network node, but which lie on a route that will **benefit** a **majority** of **packeteers** by providing **a backbone** service.

For example, there is a sufficient level of activity in Dallas, Little Rock, Oklahoma City and St. Louis to support network nodes, but insufficient activity between these regions to support linking them together. Yet, if they were linked, a major segment of a transcontinental backbone would be in place. Thus, it is to the long-term benefit of packeteers in, say, Florida (for example) to see this link established. A mechanism to fund such development should be examined. TAPR is currently looking at some schemes to accomplish this.

The point here is that parochialism must be set aside for the long-term benefits of all concerned.

6) As technical development continues, the slower-speed systems (9.6 kbps to 56 kbps) used for linking can be replaced by higher-speed nodes, with the retired equipment pressed into feeder service in the larger metropolitan nodes. This suggest3 that a form of contribution to the national effort may be as equipment "loans" for temgorsary service in remote areas.

Assuming such a scenario or one broadly similar, becomes fact: the near-term technical goals that appear achievable are:

1) Development of an integrated 9.6 kbps, low-cost 220 MHz radio/modem. Such a system is currently under active development by TAPR, in coordination with regional groups across the US. Expect something to be in the testing stage *during* the summer of 1985, with general availability as soon as testing is *'completed" (is any technical project really completed?).

2) Development of a multi-ported Network Node Controller capable of establishing Levels Two, Three and some subset of Four. A minimum capability of two ports (for a remote Location that is simply part of a backbone> and a possible expansion to as many as eight ports may be a reasonable goal.

Mike Brock, WB6HHV, has done extensive research and development of a multiple Z80-based system capable of reaching these goals. This project has been put on temporary hold pending the outcome of a decision for a Networking Protocol.

With the many new microprocessors now becoming available, and for which extensive development tools exist, this design will be undoubtedly be revised in the coming months. Again, TAPR has identified such a controller as 8 high-priority project, to be worked on in parallel with rf and modem advances.

On the non-technical front, newcomers to the packet field must be encouraged and instructed in the proper operation of Amateur packet radio. Operating procedures must be tailored to the environment as it exists today.

For example, it is a waste of channel resources to have all stations turn on their CW ID function on VHF. Beacons every few minutes telling the world you will be out of town for the next two weeks similarly have no place. Dumping 100k bytes of files during prime operating *hours* on an otherwise busy channel is similarly hard to justify.

While such abuses of common operating courtesy may seem absurd, these and other practices like them occur all too frequently on our existing, embryonic networks.

We must all work together to educate newcomers and encourage the use of proper packet operating procedure.

Conclusion

1984 was a year of tremendous growth in Amateur packet radio. 1985 has started off with even more promise of growth, injecting into our ranks a large number of non-technically oriented Amateurs.

Network protocols must be agreed upon and implemented. Hardware to support wider bandwidths, especially in the areas of rf decks and modems, must be designed and made easily available.

A coherent plan to establish regional and an eventual first national network must be determined and implemented.

Parochial interests must be moderated with recognition of the needs of the packet community as a whole.

Growth must be planned for, newcomers educated and proper packet procedures encouraged for the maximum benefit of all packetesrs while we develop the technical resources to handle our exploding operational requirements.

PACKET RADIO AND THE NATIONAL HURRICANE CENTER

Joel I. Kandel, KI4T
5463 S.W. 92nd Avenue
Miami, FL 33165
(305) 596-9373

Abstract

Amateur radio operators in South Florida are exploring the capability of Packet Radio to provide the National Hurricane Center with high quality weather observational data, and in turn transmit timely forecast information from the Hurricane Center to effected areas.

Preliminary Tests

On September 10, 1984, W4SS, ARRL Section Emergency Coordinator for South Florida called for a Packet Radio test between and among a number of county Emergency Operations Centers (EOC's).

On that date, a packet station was taken to the National Hurricane Center by KI4T, and an authentic weather bulletin was transmitted to the various EOC's participating in the test.

The weather bulletin text was received intact as far away as Melbourne, Florida, by N2WX, a distance of 220 miles, as well as by packet stations in Dade and Broward Counties, West Palm Beach, Stuart, and Ft. Pierce. This initial test of packet weather bulletin transmission from the Hurricane Center was in large measure responsible for some of the above mentioned municipalities purchasing packet controllers and dedicating computers to them.

Although the text was manually entered on a terminal at the Hurricane Center, it portents the day when automatic weather text transmissions become a standard service provided by the amatuer radio community.

Background

In order to better understand how packet radio might function superbly at the NHC, let's first look at how the NHC itself functions.

The Center is operated under the jurisdiction of the U.S. Department of Commerce National Oceanographic and Atmospheric Administration. Its function is to monitor the formation of hurricanes in the Atlantic, track their progress, and issue current weather information in the form of advisories, bulletins, updates, and forecasts to the effected areas and to the country at large. Incoming data, at the average rate of 4,000 pieces or "products" per 24 hour period, come from four major sources: (1) hurricane hunter aircraft, (2) weather satellites, (3) ships, (4) bouys, and (5) local land observations.

Hurricane hunter aircraft are most useful for meeting the hurricane out where it is forming, over water. The aircraft can send back many kinds of data as they fly directly into the eye of the storm and take some very sophisticated measurements. Like all aircraft, what goes up must eventually come down. The plane can't stay indefinitely, so its observation time is limited.

Weather satellites, specifically the GEOS series, are set in geo-synchronous orbit, 23,400 miles above Earth, so that they are stationary with respect to the Earth's rotation. While these satellites have the capability of photographing a hurricane from that position on a constant basis, it is literally an "overview" and tells little about what is happening on the ground. Last hurricane season the East Coast GEOS satellite malfunctioned. NOAA must now time share one satellite between the East Coast and the West Coast of the United States.

Ships' observations are collected in Washington and are transmitted via land-line to the NHC. While it is a source of good, first hand observational data, their positioning and presence is purely random, and there is no way of knowing when or whether any ships at all will be present in a given weather area.

Buoys located in the Gulf of Mexico and along the Atlantic Coast send back weather data through satellite. Their data is exacting, but is strictly coastal. No buoys are located in international or foreign waters of the Caribbean.

Local observations are still a most important tool for forecasting movement of the storm. Local weather bureaus abound throughout the country and the Caribbean, but are far fewer in number than amateur radio stations. With minimal packet radio equipment, amateur stations can send vital weather data back to the Hurricane Center for processing.

Amateur Radio at the NHC

For the past six years, amateur radio has played an important part in the gathering of real-time weather data for the Hurricane Center. I emphasize gathering rather than dissemination because the original and primary reason for an amateur station at the Center was to bring data into the center, rather than release information from the Center. Most are familiar with NOAA Weather Radio, broadcasting from locations throughout the country on the VHF frequencies of 162.40, 162.475, and 162.550 MHz. Fewer are familiar with the national AFOS Network. AFOS is an acronym for Automated Field Operations Services, and consists of a nationwide network of meteorlogical offices, connected to one another by a telco network of data lines and microwave relay stations.

While the AFOS system provides a high speed, 4800 baud link among the nation's weather facili-

ties in good weather, the system quickly degrades in severe weather, with flooded phone lines and micro-wave path loss resulting in the effected area being out of touch when they need the system most.

The Caribbean Islands are less equipped to keep in meteorlogical touch with the U.S., and information paths degrade even more frequently and easily than in areas of the United States proper.

It is for these reasons that the NHC has a permanent amateur radio station, donated by the Dade County Amateur Radio Public Service Corps (ARPSC). The station is manned on a 24 hour basis when the Center requests information from an effected area.

Operating Procedure

Amateur operators at NHC usually monitor the Hurricane Watch Net on 14.325 MHz., and two meter simplex frequency. Amateur operators in the community fan out across 40 and 80 meters, checking into emergency nets operating from the effected area such as the Gulf Coast Hurricane Emergency Net.

As many as 125 stations distributed from Mexico to the Florida Panhandle may check in during a given hurricane in the Gulf. These stations are asked to supply the Miami station with meteorological data such as rainfall amount, wind direction, and wind speed. These data, once received by the Miami station, are relayed to the NHC on the two meter simplex frequency.

The weather data is given to the forecasters who feed it into their main computer, which contains data from all previous hurricanes, as well as incoming data from the other aforementioned sources.

The computer develops a predictive behavioral model of the hurricane from the data, and produces a forecast of the future path of the storm. The forecast is sent out over the AFOS network, NOAA Weather Radio, and also 20 meters, by the amateur operator on duty.

For many Caribbean Islands, the amateur bulletin is the only forecast information they will ever receive.

In 1979, during Hurricane David, Dominica lost its entire weather station at the national airport. It blew away just five minutes after the amateur operator there announced that he was evacuating.

In 1980, during Hurricane Allen, St. Lucia was totally devastated, and the Prime Minister directed the entire relief effort from the only amateur radio station on the island.

The Role of Packet Radio

Packet radio has the potential to make both gathering and dissemination of timely weather information to and from the Hurricane Center infinitely more efficient, for a number of reasons.

As Bob Neben, K9BL (Rinaldo 1984: 79-82) pointed out in his paper on Packet Radio and Emergency Communications, the great advantage a packet network has over a directed phone net or C.W. net is that it can take a "bus" configuration.

This means that any station can transmit or receive on the same channel at any time, without the necessity of a net control station and without interupting the communications of other stations on that same channel. Packet radio will allow incoming weather information from amateur stations in an effected area to be transmitted on one frequency, essentially at the same time, with minimal interference.

Speed

While previous data had to be gathered laboriously on 40 and 80 meter phone through tortuous static crashes and QRM, packet promises to drastically speed up the flow of information.

Currently, the standard rate of transmission used on two meter packet frequencies is 1200 baud, or the equivalent of 1200 words per minute! This rate is nowhere as fast as current equipment allows though the FCC requires a maximum bandwidth on two meters of 20 KHz.

Still, 1200 words per minute is roughly nine times faster than normal conversational speech, and probably 18-20 times faster than passing phone traffic at writing speed; especially traffic that contains numerical statistics, passed on 80 meters,at night,through bad weather cells.

Accuracy

Since packet radio has built into it the ability to detect errors in reception, it is possible to send and receive perfect copy, an absolute must with weather data.

Man-Hour Efficiency

Since the Hurricane is manned on a 24 hour basis making manpower a premium commodity, the weather data would be much more accessible to the forecasters if received automatically. The packet station at the Hurricane Center will include a printer, allowing forecasters to access incoming data without having to disturb the amateur operator, who may be busy on a phone frequency.

If a hurricane is headed toward the South Florida area, amateur communicators are at a premium, manning shelters, municipal facilities, and other volunteer agencies. The key word here is automation.

Incoming Weather Data

Plans are currently in process to gather weather data by automated means, digitizing it, buffering it, and transmitting it back to the Hurricane Center. The Heathkit ID-4001 Digital Weather Center will be used in these tests, since it contains computer interface capabilities through a built-in 25 bit bus.

Data can be transmitted back on a regular basis, or upon the NHC packet station connecting to the remote weather stations on a rotational basis, and querrying them individually.

It is forseeable that the AFOS weather network could eventually be augmented by amateur stations throughout the country, tying into it at various nodes, and adding support to other major weather offices, beside NHC.

Outgoing Weather Forecasts

From preliminary discussions with the tech-

nical personnel at the Hurricane Center, tests will be conducted in the near future interfacing the NHC AFOS terminal to packet radio.

The AFOS data is transmitted to a number of subscribers at a large variety of baud rates, taking into account the various types and vintages of terminals in use at the other end. Approximately 500 transmissions of weather products are made from the Center each day, to civilian, marine, and military end users.

A microprocessor will have to be able to recognize a given weather bulletin heading, in order to select out the advisories pertinent to the effected area, and destined for civilian consumption. Examples of actual weather data texts and their headings are given below in the addendum.

Once these bulletins are culled from the mass, they can then be transmitted over packet radio to the amateur population at large who can then disseminate it to the appropriate authorities.

It is anticipated that these bulletins would be sent out on both HF and VHF bands, to insure that it gets to the effected areas which may be out of VHF range.

Work is currently under way to install Gator I, 9600 baud level three intercity packet link throughout Florida. High speed links of this type will facilitate the routing of weather packets.

The enclosed map indicates four major packet paths of primary concern to the Hurricane Center. These are the Atlantic Coast, the Gulf Coast, the Florida Keys, and the Caribbean. If one of these had to be singled out as being most critical it would be the Caribbean, where adequate weather communications are almost nonexistent.

The amateur radio antennas at the Hurricane Center are about 150 feet high, and offer excellent simplex two meter communications to the surrounding areas. It is anticipated that a good HF lind will be needed to both the western end of the Gulf Coast, and the Caribbean, with gateway stations in those locations able to translate HF packet to VHF, and vice-versa.

Practical, Non-technical Considerations

The NHC is a federal agency. As such, its communications are regulated not by the FCC but by the IRAC, the Interdepartmental Radio Advisory Committee. IRAC is composed of representatives from all the federal agencies, including the military and the FCC, and would have to give its consent to the transmission of AFOS information on the Amateur Radio Service. Paragraph 97.113 of the FCC regulations prohibits the rebroadcasting of information on amateur radio frequencies, though the FCC has granted a limited number of Special Temporary Authority (STA) authorizations for the transmission of NOAA weather bulletins over two meters.

An STA application may be filed previous to conducting these tests.

It remains a grey area within the FCC amateur regulations with respect to the status of packet digipeaters as remote stations, as repeaters, or even as beacon stations. The FCC should clarify their statusinthe near future in response to petitions now before it.

In any case, the operation of these stations will usually occur in an emergency situation

In any case, the operation of these stations will usually occur in an emergency situation, when there exists "... the immediate safety of life... or the immediate protection of property." Under these circumstances, the operation of a packet weather network is very feasible.

Future Technical Considerations

Much remains to be done form a technical standpoint. While digipeaters are springing up nicely throughout South Florida, not all of them are capable of running on emergency backup power, emergency power is essential to insure reliability of pathways.

Packet stations are beginning to infiltrate the Florida Keys, a crucial communications area in times of severe weather. But as of yet we do not have an established path to Key West, the southern most key as well as the most densely populated one.

Experimentation with packet on HF SSB frequencies is proceeding along, but few stations are active on HF packet, crucial to the packet weather net.

Software command parameters have to be established on the Tucson Amateur Packet Radio Terminal Node Controller (TNC), the predominant unit used in the South Florida area, to optimize the flow of data in both directions, as well as to configure the network. And last but not least, amateur stations have to be recruited to actively participate in the collection of weather data, and its transmission to the Hurricane Center.

Packet must be introduced into the Caribbean, and gateway stations established with reliable emergency power capability.

Summary

We have looked at a unique application for amateur packet radio, that of supplying crucial weather information to the National Hurricane Center from severe weather areas, and reciprocally, supplying effected areas with critical forecast bulletins when traditional sources of weather information fail.

Although not every community has a Hurricane Center in the neighborhood, a comparable weather network can provide backup information to any weather bureau office, in fact all along the AFOS circuit. The sheer numbers of amateur stations potentially equipped to do so is enormous as compared with traditional sources of weather data.

Packet radio allows for this information to be transmitted error free, with little or no loss of information over multiple, simultaneous paths, and with a high degree of speed and automation.

This type of "high-tech" amateur radio participation can only strengthen our image in the community and insure the continuity of our hobby.

Acknowledgement

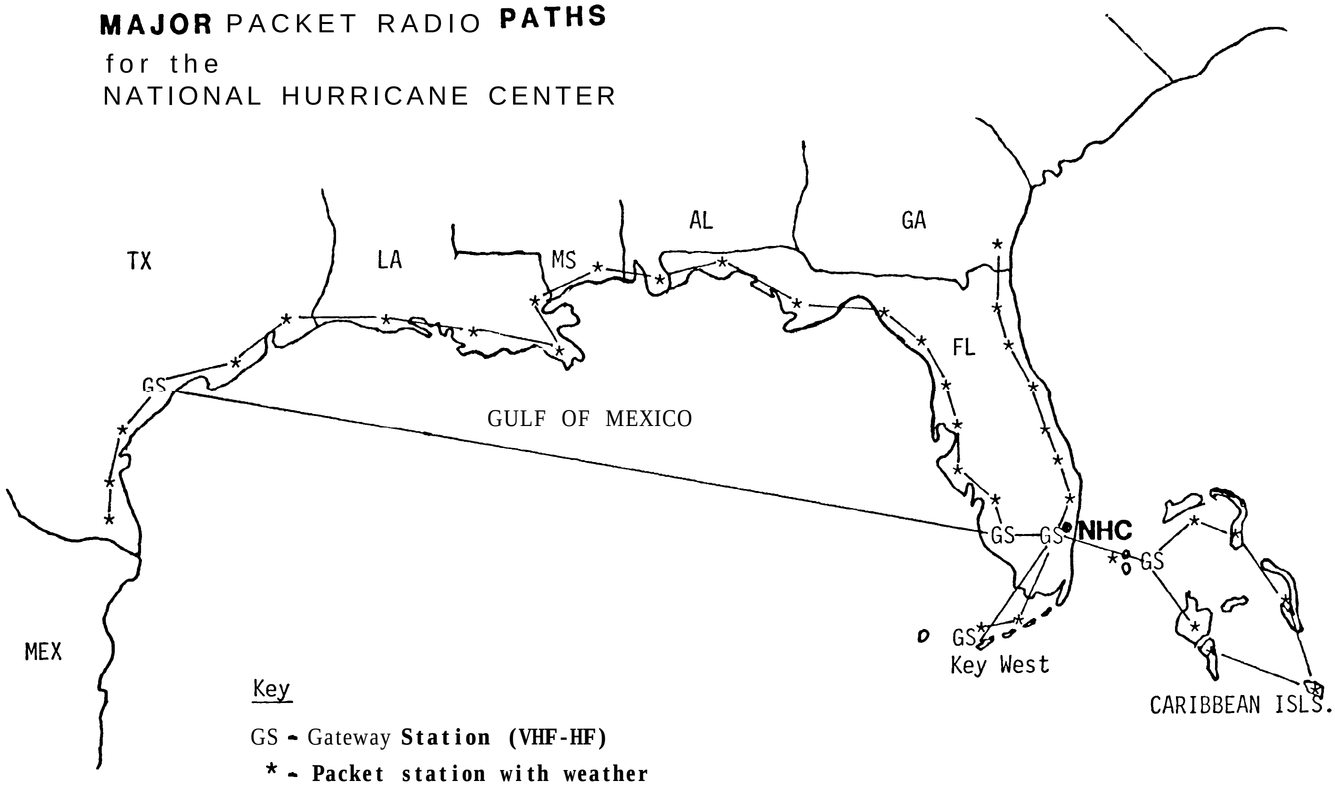
I wish to thank Dr. Neil Frank, Director, NHC, and William Pettyplace, NHC computer technician, for their cooperation and assistance; and KF4MI for typing this paper.

Reference

1984 Rinaldo, Paul L., Editor. Third ARRL Amateur Radio Networking Conference. Proceedings. Trenton.

"WETNET"

MAJOR PACKET RADIO **PATHS**
for the
NATIONAL HURRICANE CENTER



Key

GS - Gateway **Station (VHF-HF)**

* - **Packet station with weather measuring capability**

NHC - National Hurricane Center, Miami

Addendum - Sample Weather Texts and Formats

Type 1: **A Ship's Weather Report in Synoptic Code.**

ZCZC **WBC656**
SMVD6 KWBC 080600 RTD
BBXX
VSBE9 **08063 99236 70316 42308 60714 10242 20224**
40182 54000 84260 22233 00260 20101 3//// 4////

Explanation: **Preamble contains addressee's call letters, ship's call letters, shore station's call letters, time and date of message transmission.**

Five digit number groups indicate, by their position in the message sequence, date and time of reading, latitude, longitude, sky cover, wind direction, wind speed, temperature in plus or minus degrees to the tenth of a degree, dewpoint, barometric pressure, ship's directional heading and speed in knots, and wave height.

Note that the total weather report, including the heading, is approximately 128 characters long, TAPR TNC default packet length. Equivalent land-**based weather observations can be sent from amateur weather stations in similar format.**

Type 2: Sample Airport Weather Report.

SA 301300
VRB SA 1251 60 SCT 280 - BKN**7** 213/51/45/1505/016
TPA SA 1250 E50 BKN 250 OVC **10** 207/55/47/1108/014

Note three letter abbreviations for indicating city (i.e., Vero Beach, Tampa),and abbreviations for cloud cover (i.e., Broken, Scattered, Overcast). Also included in information are day, time, cloud cover elevation, configuration, temperature, barometric pressure, dew point, wind (in degrees and knots), precipitous accumulation in inches, etc.

Each line is a complete weather report, and consists of only 49 characters including spaces.

Type 3: Plain Text Reports for NOAA Distribution.

RCV 30939 14:54 01/30/85

ZCZC MIACWFMIA
WOUSOO KMIA 301500

MARINE FORECAST FOR FLORIDA AND GEORGIA COASTAL **WATERS**
NATIONAL **WEATHER** SERVICE MIAMI FL
1025 AM EST **WED JAN 30** 1985

SYNOPSIS
HIGH PRESSURE RIDGE OFF THE GEORGIA AND FLORIDA ATLANTIC COAST WILL MOVE EASTWARD OVER THE NORTH PART AS THE SOUTH PART ROTATES THROUGH THE LOWER FLORIDA STRAITS......

NNNN

Note the four N's for standard teletype printer shutdown.

TCP/IP: A Proposal For Amateur Packet Radio Levels 3 and 4

Phil Karn, KA9Q

Radio Amateur Satellite Corporation

ABSTRACT

This paper presents a case for basing Level 3 (the network layer) of Amateur Packet Radio on the "datagram" concept. It further proposes that the DARPA protocols IP (Internet Protocol) and TCP (Transmission Control Protocol) be adopted intact as the standard Level 3 (Network) and Level 4 (Transport) protocols for Amateur Packet Radio.

I will then provide an overview of TCP/IP, explain why it, as a datagram protocol, is more suitable for our needs than the virtual-circuit protocol CCITT X.75, and show how it would be used above the AX.25 Level 2 protocol already in use.

1. Datagrams and Virtual Circuits

A fundamental characteristic of ARPA (and several others, e.g., Xerox PUP [15]) protocols is the choice of the "datagram" as the fundamental unit of communication within the network. To understand what this means, a comparison of the datagram approach with its chief rival, the "virtual circuit," is needed.

1.1 What is a Datagram?

The word "datagram" is coined from the words "data" and "telegram." Like telegrams, datagrams are simple one-shot messages; each is self-contained in that it includes the full source and destination addresses, control information and user data. Each datagram is independently processed by the network. All information needed by a packet switch to route datagrams through the network is wholly contained within each datagram. No state need be maintained by a packet switch between datagrams. There are many analogies to this mode of operation besides telegrams: mailing a letter, sending electronic mail, or entering a message into the amateur radio National Traffic System.

The network makes a "best effort" attempt to deliver each datagram. If datagram delivery is impossible (e.g., due to network congestion, buffer overflow or an unknown or unreachable destination address), a packet switch may discard a datagram. Some datagram protocols (such as IP, to be described later) require that an effort be made to notify the sender of the problem.

Datagrams are never discarded lightly; however, there are usually varying degrees of "best effort" that can be expended before "giving up" on a datagram. Frequently, a greater effort at reliable delivery increases the "cost" (in some sense) of sending the datagram, or affects the user in some other way, e.g., by decreasing throughput or increasing delay. IP, as discussed later, gives the sender the ability, if desired, to specify the importance (i.e., the precedence) of a datagram and to influence any tradeoffs between delay, reliability and throughput that might exist in individual links and gateways within the network.

In any event, a datagram user must always be prepared to cope with the occasional loss, out-of-sequence delivery or duplication of datagrams caused by network congestion or switch or link failure. Since many applications require guaranteed service, a separate, higher level protocol using end-to-end acknowledgements and retransmission of lost datagrams is generally used "on top" of the unguaranteed datagram service.

1.2 And In The Other Corner... The Virtual Circuit

As the name implies, "virtual circuit" networks (hereafter abbreviated "VC networks") are oriented to provide the appearance of a direct connection between a pair of users. The network sets up a fixed path though the network to the destination for the duration of the user's connection. Because the path may be shared by several users (i.e., the physical facilities are not dedicated to a single user), the connection is "virtual."

A special "call setup" packet propagates through the network, and each switch adds a "virtual call" to an internal table so that the data packets that follow may be correctly routed to their destinations. The best analogy to a VC network is the telephone system, although the analogy isn't perfect because the telephone network usually dedicates fixed physical resources (a wire pair or a channel on an RF carrier) to each call.

Routing in most VC networks is static; once it is established at setup time, all packets follow the same route to the destination. As long as all links and switches traversed by the virtual call remain functional, the users' data will be properly delivered in sequence. However, should a switch crash or a link fail, all virtual circuits using the affected switch or link will be dropped and any data in transit will be lost.

When the user is done with a virtual circuit, it is cleared. This removes the information about the call from the memory of each packet switch along the call's path.

1.3 Discussion: Datagrams Vs. Virtual Circuits

Many applications, such as remote terminal access to a computer, require a reliable, flow-controlled "stream connection" between two end points, regardless of how this might be implemented in the bowels of the network. Therefore, the issue is NOT *whether* the user should be provided with a reliable end-to-end stream, but rather *how* it ought to be implemented. Should the concept of a "virtual circuit" be confined to the endpoints of a "connection" or should it permeate the design of the lower levels of the network?

The choice has many implications for reliability, flexibility, ease of implementation, efficiency, and adaptability to varying user-level service requirements. The decision is a tradeoff, and often the choice depends on those characteristics considered most important. Neither approach is always superior.

1.3.1 Ease of Implementation Datagram packet switches are considerably easier to implement than VC switches. The lack of special "call setup" and "call clearing" packets means that all packets are alike as far as the switch is concerned. All that it has to do is select an outgoing link for the packet (typically based on a routing table that is periodically updated from its neighbors) and send the packet on its way. If there is a serious problem with the packet, the switch is entitled to discard it; no intricate error-recovery procedures are needed. Since the "what to do when things go wrong" section is the largest, most difficult to write and least reliable section of almost any programming project, this results in an enormously easier coding job.

1.3.2 Dynamic Routing As already mentioned, virtual circuit networks establish fixed paths through a network of packet switches and links. If a given link fails or becomes overly congested, there is no easy way to re-route established virtual circuits via alternate paths.

Datagrams, with their self-contained nature, may be individually routed without regard to any end-to-end connections that might exist at a higher protocol level. This makes it possible to react on a per-packet basis to changing traffic conditions and network reconfigurations. Much of the work to date in non-amateur packet radio has been done in a mobile environment, and dynamic routing is essential here because of the constantly changing topology of the network.

While it is certainly possible to make routing decisions based on link loading at circuit setup time in a virtual circuit network, this is less responsive to rapidly changing network conditions than the ability to route on a per-packet basis.

1.3.3 Overhead This is *the* primary objection that is made against datagram protocols. Virtual circuit protocols require that complete addresses be sent only at circuit setup time. Once the table entries are made in each switch along the path of a virtual circuit, only the index into this table (referred to as a "virtual circuit number") need be part of each data packet for the switch to route it properly. Depending on the size of the data fields, the larger headers involved in datagram packets can involve considerable overhead. This is primarily true with interactive terminal traffic that often consists of single character packets; it is much less of a factor when data fields are larger.

On the face of it, virtual circuit protocols seem to win the overhead argument hands down. However, there are applications where the direct availability of a datagram service to the user (e.g., the ARPA User Datagram Protocol, UDP [11]) results in fewer packets and bits being exchanged to accomplish the same task.

Such applications typically have a "client-server" characteristic. For example, a database server might be set up to provide "directory information" (i.e., providing the network number corresponding to a given station's name). Most transactions with such a database server are short; the request and replies each fit easily into single datagrams. In a virtual circuit network, a virtual circuit must be first set up between the client and the server, the request made, the response received, and the virtual circuit torn down. This clearly results in more network traffic than if one-shot datagrams were used at the network level, avoiding the overhead of setting up a virtual circuit for such a short "connection."

To answer this objection, the X.25/X.75 protocols include an optional "fast select" feature that allows user data to be sent in the same packet with a call request. Fast select is not, however, a substitute for datagrams. A virtual circuit is still being established, although for a short time. A reply packet (typically a CLEAR INDICATE), with or without data, is still expected from the destination within a time limit imposed by the network, and the data fields contained in either packet are limited to 128 bytes; there is no fragmentation facility. This is considerably less general than a "true" datagram facility.

However, in the common situation where the application requires an end-to-end connection for a relatively long time, virtual circuit networks do require fewer bits to be transmitted than do datagram-based networks. In situations where the traffic consists primarily of single-character packets (e.g., interactive terminal access) and efficient use of slow and expensive transmission facilities is of supreme importance, the lower per-packet overhead of the virtual circuit approach can be the overriding factor. [13]

While amateur packet radio is currently severely constrained by obsolete Bell 202 modems and 1200 baud transmission, dedicated RF modems operating at much higher speeds (orders of magnitude) are now being introduced. [16] Since these modems will not cost much more than those currently used (assuming a dedicated radio), this will almost completely mitigate the overhead argument.

1.3.4 Reliability Because a datagram contains all information necessary to forward it onto its destination, no state has to be maintained in a datagram packet switch between packets. This makes datagram networks much more resistant to real-world occurrences such as power glitches, software failures and nosy visitors who push reset buttons.

Since the reliability requirements are less for a datagram switch (since it has no volatile table of virtual circuits to safeguard) such measures as battery backup can often be dispensed with.[1] The only information that is typically lost within datagram packet switches during failures are routing connectivity tables (assuming a distributed routing algorithm is used), but these can be quickly rebuilt from one's neighbors. Virtual circuit switches, on the other hand, must maintain the information provided to it at circuit setup time to route successfully each data packet of a virtual connection. In general, this information cannot be rebuilt from one's neighbors, and the end user must re-establish the virtual circuit and recover from any lost data.

To achieve maximum reliability against internal network problems, both datagram and virtual circuit networks require a higher-level end-to-end "transport" protocol. A transport protocol recovers from various errors that might occur in the network (lost, reordered or duplicated packets in a datagram network, or dropped virtual circuits in a virtual circuit network). The transport protocol used atop the ARPA datagram protocol, IP, when reliable stream communication is desired is called TCP (Transmission Control Protocol).

A major advantage of an end-to-end protocol such as TCP is that it provides protection against data corruption (as well as loss) along the ENTIRE network path. Link level error detecting codes (such as the 16-bit CRC in AX.25) protect only against errors on transmission links. Without end-to-end protection, a user is still vulnerable to data corruption that can occur in a packet switch between the reception of a packet and its retransmission with a freshly regenerated CRC. The probability of this occurring in a single packet switch may be acceptably small, but in a large network composed primarily of inexpensive microcomputers without memory error detection, errors are inevitable.

Many virtual-circuit proponents claim that their networks provide "reliable" VC service with less complexity than datagram networks because they do not "need" an elaborate end-to-end transport protocol. However, many X.75 networks provide NO end-to-end transport protocol at all, and as a result the user is still vulnerable to failures within the network that can lose information or drop connections. In practice, this happens often enough to be annoying. With an end-to-end transport protocol on a VC network, the reliability can approach that of, say, a TCP/IP network, but now the total implementation complexity is greater because of the redundancy at multiple levels.

1.3.5 Grades of Service VC networks are implicitly based on the assumption that *all* applications require a reliable, flow controlled stream "connection." However, there are several real-time[2] applications that either do not require

1. Most power failures are very brief, and if the switch recovers within the abort timeout interval of the end-to-end protocol (or if an alternate route is available), the only effect may be a momentary "freeze" on data transfer, not a dropped connection.

2. "Real time" as used here means that the information being transmitted is useful only for a short time until a

this grade of service or cannot tolerate any overhead introduced by it.

The best example of an application in this category is packet voice. People conversing on a telephone channel are sensitive to long transmission delays, especially if they are irregular. In contrast to data transmission, however, human speech can tolerate a certain amount of lost or corrupted data because of its great redundancy, and "perfect" reliability may be sacrificed to reduce delay.

Other examples of real-time applications might include television ("digital SSTV") and satellite telemetry. In each case, there is little point in retransmitting lost "old" data because "new" information will arrive shortly to take its place. For example, real time satellite telemetry gains little from retransmission of lost frames; the user might as well wait for updated information (and then interpolate the missing values) instead of falling behind by trying to recover data that is already out of date. If a somewhat higher degree of reliability is needed (but the cost of "perfect" reception is too high) the satellite might simply repeat each frame of data several times to increase the chances of successful reception, or use other more complex forms of forward error correction (FEC).

In a datagram network, these kinds of application-specific tradeoffs are easy. For example, control bits in each datagram might select the use of hop-by-hop acknowledgements. In a VC network, however, all applications get hop-by-hop acknowledgments whether they need them or not. Other applications might place different levels of importance on different messages (e.g., emergency vs routine traffic) but because VC networks typically handle traffic on established virtual circuits on a first-come, first-served basis this is difficult to do.

While it may be a while before amateur packet radio networks have the capacity to handle packet voice at a practical level, it would be unwise and shortsighted to adopt a protocol that would effectively *preclude* it from our network. Much could be gained through the joining of resources that could occur if a common network could satisfy the needs of amateur data *and* voice users.

1.3.6 Broadcasting Virtual circuits are inherently point-to-point and usually full-duplex, and thus they do not lend themselves easily to the notion of a "broadcast" message. Sending the same information to N receivers requires that N virtual circuits be created, one to each receiver, and that N copies of the data be transmitted. This is clearly wasteful when the underlying media permits broadcasting (such as Ethernet [10] or radio), and datagrams are a much more natural solution.

Given that reliable delivery to every receiver in a broadcast environment is much more expensive than reliable delivery to a single destination, it is even more appropriate to provide an unguaranteed service. As with simple point-to-point connections, a reliability-improving mechanism appropriate for the specific application would then be implemented on top of basic broadcast datagrams.

Other situations where broadcast mechanisms are useful include the construction and exchange of routing information, and in distributed processing to manage a collection of systems providing a set of services.

As with dynamic routing, datagrams do not, in themselves, solve every problem involved in broadcasting, but they do not *preclude* it outright as do virtual circuit networks.

"deadline" is reached. Information arriving after the deadline is useless, even if it is received correctly.

2. What is TCP/IP?

The ARPA Transmission Control Protocol (TCP) [3] and the ARPA Internet Protocol (IP) [1] are part of a larger collection of protocols that enjoy widespread and rapidly growing usage within numerous commercial, research and military computer networks.

Before delving into the internals of these two protocols, it is necessary to understand the needs of their developers and environment where they were designed.

The ARPA research community that designed TCP/IP is a *user* (as opposed to a *vendor*) of computer hardware and communications facilities, and this had a major impact on its design. [9] A basic requirement was the interconnection of many dissimilar types of computers, using the widest possible variety of link-level networking hardware and protocols. This was important for two reasons: first, much of the hardware already existed and couldn't be thrown away. Second, the user community wanted a "hardware independent" protocol to guard against becoming "locked in" to any one vendor's products. This is in contrast to a vendor's usual incentive to establish standards that favor the use of ones' own products over those of the competition.

It was found that the networks in use vary radically in their characteristics. Some support the concept of "connections"; others only provide unguaranteed delivery. All vary widely in reliability, transmission speeds and addressing formats. The datagram was the only feasible choice as the "common unit" of transmission that could be "encapsulated" on each of these heterogeneous networks.

Other ARPA requirements included robustness in the face of internal network failures and reconfigurations, provisions for precedence, class-of-service and security classification, and optional user specified routing. Because no existing protocols satisfied these requirements (including CCITT X.25/X.75), it was necessary to design a new set of protocols.

The widespread acceptance of TCP/IP outside the military community that originally sponsored its design shows its success in meeting the needs of a wide variety of users, not just those of the military. The latest available figures show that address assignments have been made to a total of over 3,000 distinct networks of varying sizes. About half of this total represent Defense and Government-sponsored research organizations that are interconnected to form the ARPA Internet, while the rest are independent private (mostly commercial) networks. While the exact total number of hosts that support the Internet protocols is unknown, the ARPA Internet host file currently contains 1,146 hosts, ranging from IBM PCs to large timesharing systems.

Planning activities within the International Standards Organization (ISO) now include the design of a protocol based on TCP/IP (TP4), although the CCITT seems to remain adamantly opposed to this type of protocol.

In the following sections, I will discuss the major features of the ARPA IP and TCP protocols. In general, the statements made earlier about datagram protocols apply to TCP/IP. In addition I will point out some differences between TCP/IP and X.25/X.75 that are specific to those protocols and not necessarily related to the datagram/virtual circuit selection.

2.1 The Internet Protocol (IP)

The Internet Protocol (IP) occupies level 3, the network layer, in the ARPA protocol "suite." As its name implies, IP is the "universal language" of the network; it is the "Esperanto" of a larger network built up through the interconnection of many smaller, heterogeneous networks.

IP is a datagram protocol that makes minimal assumptions about the links it uses. IP headers contain only that information necessary to provide network functions such as addressing, classes of service, precedence, etc. In particular, there are no end-to-end features such as guaranteed delivery, flow control, sequencing, or other services commonly found in virtual circuit protocols. As a result, IP is simple and easy to implement on a wide variety of networks, including many that cannot directly support virtual circuits. Intermediate relay points (hereafter called "gateways") only need implement IP in addition to whatever link level protocols are being used; any end-to-end functions remain the domain of higher level protocols in the user systems.

The maximum size of an IP datagram is 65,536 bytes. Since many (most?) networks cannot handle such large packets, IP provides a feature called *fragmentation*. This allows a gateway faced with a datagram that won't "fit" into a given link level protocol to split it into several smaller datagrams that will. Each "fragment" behaves like a separate datagram in its own right and will propagate independently through the network. Only when they arrive at their destination will they be "reassembled" into the original, larger datagram and passed to the next layer protocol. As we will see later, this is an important feature that eases the use of IP on top of AX.25 Level 2. [12]

Each IP datagram contains a "Time To Live" (TTL) field that is decremented as the datagram propagates through the network. If the TTL reaches zero before the datagram is delivered, it is destroyed. Of course, the TTL field is set to a large enough value so that the datagram is likely to reach its destination; however, if a transient routing loop occurs in the network the datagram will not circulate indefinitely. Even if the routing loop eventually disappears, the TTL field protects higher level protocols by establishing the maximum interval when they must guard against duplicates of a particular datagram. The TTL feature provides backup protection against buffer deadlock by ensuring that a datagram never remains in the network indefinitely. A colorful (although admittedly impractical) analogy might be the placing of a time bomb in each car entering New York City. Normally, the cars leave the city in plenty of time, and there are few (or no) explosions. If gridlock occurs, however, even in the absence of any other actions taken the problem is guaranteed to go away eventually by itself!

Several features of IP are not used frequently enough to justify their inclusion in every datagram. These "IP options" are listed in [2], but the most interesting ones include:

1. Source routing. Normally, gateways do their own routing, but two forms of user ("source") specified routing are available. One, "strict source routing" specifies the exact path that must be used by the datagram; if this path is invalid, the datagram is discarded and a failure report is sent to the user. The other form, "loose source routing" allows the user to only partially specify the route; the gateways are free to determine paths between each user-specified point.

2. The "record route" option asks the gateways to route the datagram automatically, but to record in the datagram the path used.

3. Related to the "record route" option is the "Internet Timestamp" option. This option requests that each gateway record the time when it processed the datagram.

Most datagrams are sent without any of these options. However, they are extremely useful for special functions such as testing or collecting statistics about specific paths within the network. It should also be pointed out that X.25/X.75 provides none of these features. If they are considered necessary for amateur packet radio they would have to be added to those protocols.

A special "protocol" called ICMP (Internet Control Message Protocol) is considered an integral part of IP. ICMP is simply a standard way for an IP gateway to send a report back to the originator of a datagram when some unrecoverable error occurs. Gateways are required to generate ICMP messages whenever a datagram must be dropped for any reason, with the sole exception that ICMP messages are never generated about other ICMP messages (to avoid endless loops). ICMP messages report such error situations as invalid IP header formats, unreachable destinations, buffer congestion, time-to-live fields expiring, etc.

Other ICMP messages are of a more advisory nature. The "Redirect" ICMP message is used to notify a host's routing algorithm that an alternate gateway is a more optimal path to a given destination. There is also an ICMP "echo/echo reply" message pair that allows a host to monitor network performance by "pinging" echo requests off selected destinations, perhaps using specified routes.

2.2 The Transmission Control Protocol (TCP)

TCP is the standard ARPA Level 4 (Transport) protocol. TCP, and not IP, provides optional end-to-end "virtual circuit" service to applications that require it. Consistent with the concept of a datagram network, TCP resides only at the endpoints of the connection (typically in the computers containing the application programs) and not in the intermediate gateways. TCP takes the (potentially) unreliable service provided by IP and provides a reliable stream. Therefore it must resequence datagrams that have been delivered out of sequence by the network, detect and discard duplicate datagrams generated by the network, and request retransmissions when data is lost altogether.

While internal details of TCP are beyond the scope of this paper (see [3] for the formal specification of TCP), several of its key features can be mentioned here.

One is that each character of data has its own sequence number. This doesn't mean that TCP sends single-character packets; TCP "segments" (i.e., the datagrams that it sends by way of IP) can be of any length up to a limit negotiated by the TCP "maximum segment size" option. However, in contrast to X.25, receiver acknowledgments show exactly how many bytes of buffer space (the "window") are available.

In X.25 level 3, sequence numbers refer to "packets" that could be of any size from 1 to, say, 256 bytes. The receiver's "vocabulary" for flow control is limited to two phrases: "receiver ready (RR)" and "receiver not ready (RNR)." How MUCH the receiver is actually ready to accept when it says "RR" must be agreed on in advance and specified to the protocols. The maximum is 7 with standard sequence numbering, while a typical value is 2 packets. This means that the receiver cannot confidently indicate that it is ready to receive any data at all unless it has at least enough buffer space for two full-size (256 byte) packets. At the other end of the circuit, the sender may send no more than two (in our typical example) packets, even if each of these packets contains only a single character. This obviously limits efficient buffer utilization, and can severely limit throughput as well.

2.3 AX.25 and Digipeaters: A Poor Man's TCP/IP?

Those who have followed the development of AX.25 Level 2, particularly the digipeater feature, may have noticed a certain peculiar similarity to the way things are done under TCP/IP.

What we call "AX.25 Level 2" is, in fact, composed of two distinct "sub-layers." The upper of these two sub-protocols is the familiar connection-oriented, end-to-end

byte stream protocol essentially identical to X.25 LAPB. However, the essential changes that were made to X.25 LAPB to form what is now known as AX.25 Level 2 consisted entirely of inserting a *datagram* layer below LAPB! Every packet contains full source and destination addresses, the touchstone of a datagram protocol. In addition, should digipeaters be used, each packet contains a "strict source route," in IP terminology.

When AX.25 is used directly between two points (i.e., no digipeaters are used) it looks reasonably like an ordinary link level protocol. However, when digipeaters are used, the virtual-circuit level of AX.25 (the transmission of connect requests, sequence numbers, etc) is "promoted" to serve as an end-to-end *transport* protocol analogous to TCP. The digipeater is, in fact, nothing more than a *datagram* based packet switch, although it is very simple because it can't do routing.

Nor would automatic routing by digipeaters be desirable, since LAPB, which was intended solely as a *link* level protocol, does not make a very good *transport* protocol. For example, it is totally confused by packets that arrive out of order or duplicated, events that inevitably occur occasionally in datagram networks with automatic routing mechanisms, but not on the point-to-point links for which it was designed. There are also situations where information can be lost in LAPB requiring recovery by higher level protocols; this is unacceptable behavior by a transport protocol, the user's last defense against data corruption.

However, AX.25 *without* digipeaters is entirely suitable as a link level mechanism for relaying IP datagrams from one packet switch to another, and it can play an important synergetic role here. A major problem with our existing ad-hoc digipeater networks is the lack of hop-by-hop acknowledgements. If a packet in transit down a chain of digipeaters is lost for any reason, the transmission must be restarted back at the source. Even more wasteful is the loss of an acknowledgement in transit, as this requires the retransmission of the data being acknowledged as well as the retransmission of the acknowledgement. If the probability of successful transfer between adjacent digipeaters were high enough, end-to-end retransmission would be rare and would not be much of a performance problem. However, many problems, including the "hidden terminal problem,"[3] poor RF links and channel overloading, cause significant numbers of packets to be lost in real digipeater networks. If IP datagrams were to be sent in "raw" HDLC frames, a long multihop TCP/IP connection would suffer as much from this problem as a long multihop AX.25 connection. However, if an AX.25 link existed between each point of a long path, with the regular acknowledgment mechanism being used to increase the chances of a packet being successfully relayed onward, the efficiency and throughput of our network would increase dramatically. The end-to-end transport functions would instead be handled by TCP, a much more robust protocol specifically intended for this job. Retransmissions by TCP would be quite rare and would occur only when a link failed or became congested within the network.

It is in this way that AX.25, as a layer 2 protocol, and TCP/IP, as layer 4 and 3 protocols, respectively, can complement each other to produce an effective multihop network. The next section deals with the details of consummating such a marriage. Much of it is modeled on an existing standard for the transmission of IP datagrams over Public Data Networks using the X.25 interface. [7]

3. Sending IP Datagrams on AX.25 Links

IP was designed to be easily "enveloped" in a wide variety of link level protocols, and the AX.25 link level is easily capable of supporting it. However, a standard must be established, and several items have to be addressed. A proposed standard is presented here; a summary is contained in Appendix A.

3.1 Protocol ID

The Layer 3 Protocol ID byte immediately follows the control field in AX.25 Level 2. Until version 2.0 of AX.25, only a single value had been defined: hex F0, meaning "no layer 3", i.e., send the packet directly to the terminal. For AX.25 Version 2.0, the Protocol ID byte hex CC has been defined to mean "Internet Protocol."

3.2 Service Mappings

Two types of frames (I and UI) in AX.25 may carry information. I-frames are sent using the full LAPB (connection-oriented) facility, while UI-frames allow access to the underlying datagram sublayer of AX.25 and do not provide guaranteed delivery. How should the "class of service" bits in the IP header control the selection of the frame type used to send the datagram?

Two "class-of-service" bits are relevant, "low delay" and "reliability." A reasonable mapping would seem to be the following: If the "low delay" bit is set (and the others are not), then send the datagram in a UI frame (i.e., do not use per-hop acknowledgements). On the other hand, if the "reliability" bit is set, then use I frames (i.e., use the per-hop acknowledgments of AX.25 level 2 to increase the chances of the datagram being successfully transferred to the next gateway). If neither (or both) of these bits are set, then it is up to the individual gateway whether to use I or UI frames. This choice may be based on local conditions, e.g., experience in the reliability of a given RF link.

It is not clear how the "throughput" bit should be interpreted, so for the time being it should be ignored. In the ARPA Internet, it is used by gateways that must choose between a low delay (but low throughput) terrestrial channel and a high throughput (but large delay) satellite channel in reaching a given destination.

3.3 Fragmentation

The AX.25 Level 2 document specifies that the maximum size of an I-field shall be 256 octets. This means that an IP datagram that is larger than 256 octets must be split into several using IP's fragmentation facility. Since hosts on the ARPANET often send 512 byte datagrams (to reduce header overhead) this facility must be incorporated if our network is to interconnect with non-AX.25 based sites.

3.4 Address Resolution

An IP address is a fixed-size 32 bit field, too small to contain an amateur callsign. Widening this field is out of the question since it would no longer be IP (remember that IP is the basic, universal protocol that is absolutely standard across a wide variety of systems). Nor would it be desirable to use amateur callsigns as IP addresses even if they did fit, but this is a topic relegated to my companion paper, "Addressing and Routing Issues In Amateur Packet Radio."

Therefore, there must be some way to map between the addresses used at the IP level and the addresses (call signs) used at the AX.25 link level. Fortunately, an almost identical problem has already been solved in the ARPA community, that of sending IP datagrams over the Ethernet local area network.

3. The hidden terminal problem in packet radio occurs when stations interfere with the reception of packets intended for other stations because they are unable to hear (and defer to) the transmission being interfered with.

Ethernet link level addresses are 6 bytes long. Unique addresses are programmed by the manufacturer into a ROM on each Ethernet controller, and they cannot be easily changed by the user. After several unsatisfactory ad-hoc kludges, a solution proposed by David Plummer of MIT was widely adopted, and it has worked well. Plummer's Address Resolution Protocol, or "ARP" [6] (not to be confused with "ARPA") has been widely adopted and is general enough to work on other broadcast-type local area networks besides Ethernet (such as packet radio).

ARP works as follows. Whenever a station needs to determine the link layer address (e.g., the Ethernet 6-byte address or the AX.25 Level 2 call sign and sub-station ID) corresponding to a given 32 bit IP address, it broadcasts a special "ARP Request" packet on the channel. The station with the requested address responds with an "ARP Reply" packet containing the desired link level address. Naturally, to avoid having to invoke ARP for every datagram each IP station maintains a cache table of IP to link address correspondences. This table does not have to be large since it will contain only those stations that are "neighbors," i.e., stations to which packets can be directly sent using level 2. Packets addressed to more distant sites will be sent first to a gateway, and it is only the gateway whose link layer address is needed. Entries in the ARP table are aged, occasionally purged, and replaced with the reply to a fresh ARP request to allow for the possibility of network reconfiguration (i.e., stations changing their IP addresses.)

The beauty of ARP is that it works automatically and transparently. The IP layer need not be concerned with link layer addresses, and there is no need to maintain manually a table of IP and link level addresses. In practice, it is even possible to swap Ethernet boards (and their addresses) between computers without any adverse consequences.

ARPA has already assigned an ARP "hardware type" value of 3 to AX.25 Level 2. ARP request and reply packets will carry AX.25 Level 3 Protocol Identifier CD (hex).

3.5 Addressing and Routing

This a major challenge facing any higher level Amateur Packet Radio protocols, virtual circuit or datagram. Since the purpose of this paper is to argue the case for TCP/IP, I have devoted a companion paper, "Addressing and Routing Issues in Amateur Packet Radio," to this topic as these issues apply equally to an IP or a X.75 network. I will simply mention here that ARPA "Class A" network number 44 has already been assigned to amateur packet radio through the foresight of Hank Magnuski, KA6M. IP addresses are always 32 bits wide; addresses within the ARPA assignment would contain 44 (decimal) in the first 8 bits of the address, and the assignment of the remaining 24 bits is left up to us. This provides the ability to address 16,777,216 individual stations.

4. Conclusions

It is difficult to summarize in just a few words what has been argued about at length by so many people, only a tiny fraction of whom are involved in amateur packet radio. Nevertheless, TCP/IP's proven flexibility and adaptability makes it an extremely attractive candidate for our needs. It already provides virtually every function we need in an amateur packet network and can be adopted exactly as-is, keeping open the possibility of direct interconnection with non-amateur TCP/IP networks.

TCP/IP is ideal for amateur radio, a heterogeneous environment where stations come and go, propagation paths change, satellites rise and set, and users experiment with new applications and transmission schemes unforeseen at present.

In contrast, X.25 and X.75 are much more limited protocols designed for a homogeneous common carrier environment with static network topologies, reliable nodes, point-to-point transmission lines, and a limited set of user services. Many ad-hoc changes would be required if they were to be used on amateur packet radio, creating new, unique and incompatible protocols. Interconnection with non-amateur networks would require protocol conversion gateways, a totally unnecessary complication.

Virtually all non-amateur packet radio systems use TCP/IP (and none whatsoever use X.75, to my knowledge). Furthermore, the fact that amateurs are already using a protocol much like TCP/IP (i.e., AX.25 with digipeaters) gives strong support to the convenience, flexibility and simplicity of this approach. If we adopt TCP/IP, we will be able to tap the enormous amount of experience that has been gained (and made public) with it over its 10+ year evolution. If instead we adapt X.75 and the higher layers of X.25, we will be forced to solve (or endure) many of its deficiencies in an ad-hoc, unique and time consuming way.

5. References

[1] Postel, J., ed., "Internet Protocol - DARPA Internet Program Protocol Specification," RFC 791, USC/Information Sciences Institute, September 1981.

[2] Postel, J., "Internet Control Message Protocol - DARPA Internet Program Protocol Specification," RFC 792, USC/Information Sciences Institute, September 1981.

[3] Postel, J., ed., "Transmission Control Protocol - DARPA Internet Program Protocol Specification," RFC 793, USC/Information Sciences Institute, September 1981.

[4] Postel, J., "Internetwork Protocol Approaches," IEEE Transactions on Communications, April 1980, page 604.

[5] Postel, J., and Reynolds, J., eds. "Assigned Numbers," RFC 923, USC/Information Sciences Institute, September 1981.

[6] Plummer, D., "An Ethernet Address Resolution Protocol or Converting Network Protocol Addresses to 48-bit Ethernet Addresses for Transmission on Ethernet Hardware," RFC 826, MIT-LCS, November 1982.

[7] Korb, John T., "A Standard for the Transmission of IP Datagrams Over Public Data Networks," RFC 877, Purdue University, September 1983.

[8] Padlipsky, M. A., "A Critique of X.25," RFC 874, Mitre Corporation, September 1982.

[9] Padlipsky, M. A., "The Illusion of Vendor Support," RFC 873, Mitre Corporation, September 1982.

[10] "The Ethernet, A Local Area Network: Data Link Layer and Physical Layer Specification," X3T51/80-50, Xerox Corporation, Stamford, CT., October 1980.

[11] Postel, J., "User Datagram Protocol," RFC 768 USC/Information Sciences Institute, August 1980.

[12] "Amateur Radio AX.25 Link Layer Protocol Version 2.0," American Radio Relay League, publication pending.

[13] L. G. Roberts, "The Evolution of Packet Switching," Proc. IEEE, November 1978, page 1307-1313.

[14] Lampson et al., eds., "Distributed Systems - Architecture and Implementation," Springer-Verlag, 1981., chapter 6. Distri

[15] ibid, chapter 19.

[16] Steve Goode, K9NG, personal communications.

6. Appendix A

6.1 Datagram Encapsulation

1. Each datagram shall be sent in a separate AX.25 Level Two I or UI frame. The IP header immediately follows the AX.25 Level 3 Protocol ID byte, which shall be set to CC hex denoting INTERNET PROTOCOL. No other headers are to be used, and the AX.25 Level Two Frame Check Sequence (FCS) immediately follows the last byte of the datagram.

2. The maximum length of an IP datagram, including the IP header, headers for higher level protocol(s), and user data, shall not exceed 256 bytes. The IP fragmentation/reassembly facility shall be used on any datagram larger than this limit. The possibility of increasing the 256 byte fragment size limit is a subject for future study.

   Gateways may choose to fragment datagrams to sizes less than 256 bytes when necessary to increase the chances of successful transfer over links with high bit error rates; however, this should be done as a last resort.

3. The choice between the use of an I or a UI frame for the transmission of a datagram is made by examination of the "Class of Service" bits in the IP header. Datagrams with the "Reliability" bit set to one must be sent via I frames over regular AX.25 Level Two connections. Datagrams with the "Low Delay" bit set to one must be sent in Unnumbered Information (UI) frames. If neither or both bits are set, then each link node may make its own choice between I and UI frames based on local considerations.

   The interpretation of the "Throughput" bit is a subject for future study; in the meantime it should be ignored.

4. Buffer space permitting, each Level Two implementation should be able to accept and process UI frames containing IP datagrams whenever they are received, whether or not a regular Level Two connection exists with the sending station or any other station.

5. AX.25 Level Two connections may be established on demand when needed to transmit datagrams with the reliability bit set, or they may be continuously maintained; this is a local option to be agreed on by the stations concerned. If an AX.25 Level Two connection is to be taken down, each station should make every effort possible to ensure that any outstanding datagrams sent via I frames (i.e., datagrams with the "reliability" bit set) have been sent and acknowledged before going into the disconnected state.

6. There is no effort to maintain Level Two connections corresponding to any end-to-end virtual circuits that may exist at higher protocol levels.

6.2 Address Resolution Protocol

Whenever a station needs to determine the AX.25 Level Two address (i.e., the amateur radio callsign) of another station in its local area corresponding to a given Internet address, it shall use the ARPA Address Resolution Protocol (ARP) as described in RFC 826. The value of the "hardware type" field in an ARP packet has been assigned by ARPA to be 3, meaning "Amateur Radio AX.25".

Each ARP broadcast and reply packet is sent in a separate UI frame immediately after the AX.25 Level 3 Protocol ID byte, which is set to CD hex denoting ADDRESS RESOLUTION.

The contents of the AX.25 Level Two destination field to be used for all broadcast packets (including ARP) shall be "QST" with SSID 0.

1 It may be occasionally necessary to use an IP gateway that cannot support the full multiple-connection-oriented AX.25 facilities due to memory limitations. Such gateways should be used only when absolutely necessary, and when the quality of the associated RF links is high enough to provide reasonable reliability without acknowledgments.

Addressing and Routing Issues in Amateur Packet Radio

Phil Karn, KA9Q

Radio Amateur Satellite Corporation

ABSTRACT

As amateur packet radio evolves from scattered, ad-hoc collections of local area digipeaters into a large, automatic, and interconnected network, several issues related to naming, addressing and routing will have to be faced and overcome.

Routing, in particular, has long been a fertile research area in computer networking. I make no claim to knowing the answers to many of these problems; however, I believe that they can at least be stated, and that certain decisions can be made early to ease experimentation with various solutions. In particular, the problem of address assignment is discussed with particular emphasis on making the routing problem easier.

1. Introduction: Terminology

I will begin by defining several important terms. A *link* is any transmission line, radio channel or the like capable of carrying a packet directly between two points. *Nodes* are the end points of links. A node may generate packets for other nodes, consume packets addressed to itself, or act as a relay point for packets originating from and addressed to other nodes.

Reference [1] gives a concise but effective definition of three more concepts: "In simple terms, a *name* tells what an object is; an *address* tells where it is; and a *route* tells how to get there."

To elaborate:

1. A *name* is an arbitrary string of characters, chosen for human convenience, to designate a particular person, node or service. Examples include people's names and the network names given to computers. Amateur radio callsigns might also qualify, although many may dispute their convenience!
2. An *address* is a number corresponding to a name, significant to a communications network. It is generally smaller than a name and has a well-defined format. Examples include telephone numbers (10 decimal digits in North America), Internet Protocol addresses (32 bits) and, of course, postal addresses.
3. A *route* is the path over which a specified address may be reached from a given point within a network.

Some communication systems blur the distinction between these concepts. For example, an old-time telephone system with a human operator might accept either a person's name or a telephone number in placing a call. A courier might accept a route ("go to the third (red) house on the right after making a left turn at the light") in place of an address. However, machines demand short, precise identifiers that are often not convenient for humans to remember, so translation from names to addresses then to routes must be provided. Telephone books and directory assistance systems provide translation between names and telephone numbers, while maps provide the information necessary to find a route to a given street address.

While names are generally arbitrary (except that they must be unique, at least in the context in which they are used), addresses may or may not imply something about physical location to facilitate route selection. For example, telephone numbers in North America are assigned by a multi-level location-dependent hierarchy; the first level is the 3-digit area code and the second level is the 3-digit central office code. If you move from New York to Los Angeles you are not allowed to keep the same telephone number; you must get a new one that reflects your new location. You are allowed to keep your name, however; the telephone company grants you this one concession!

Because of the arbitrary nature of names, full-blown database systems are generally required to convert names into addresses. The telephone network provides name-to-address mapping through telephone books and directory assistance bureaus. Computer networks may use either or both of these techniques, e.g., a directory may be maintained locally in each user system, or a central site may be set up as a "name server."

2. Naming In Computer Networks

The main problem here is in verifying the uniqueness of a selected name. For small networks, a single central clearinghouse for name assignments is practical. For large networks, however, hierarchical allocation is the most practical approach.

The ARPA Internet has grown rapidly over the past 5 years, and its central name registry is showing considerable signs of strain. To answer this problem, the ARPA Internet will be evolving from a single, globally administered name space to a hierarchical "domain-based" system [12,13]. In the domain system, a name may consist of several words separated by periods: FOOBAR.ARPA, meaning host name "FOOBAR" under domain "ARPA." The same system might be more fully named as FOOBAR.MIT.ARPA, distinguishing it from any other hosts that might also be named "FOOBAR" in other locations within the ARPA "top-level domain." Partially specified domains are interpreted based on the location in the hierarchy where the name is encountered, and "nearby" (in a hierarchical sense) hosts might be named without any domain names at all. This is analogous to the way people refer to others by their first names (e.g., within a family or work group), but would give full names when referring to someone on the "outside."

The ARPA Internet has only begun its conversion to the domain system, and several details need to be worked out. Within amateur radio, the easiest naming convention would simply be to use our callsigns, since they are already unique. A domain name could be allocated (e.g., AMPRNET) so that when the ARPA Internet finishes its conversion our callsign-names would simply become, e.g., "KA9Q.AMPRNET" outside our network.

3. The Routing Problem

Once the network is given an address, it must select a route to reach it. Again, this may be done centrally, in a network manager that keeps track of the entire network (e.g., TYMNET [4]), or it may be done in a distributed fashion by local routing algorithms that operate from partial, locally constructed views of the network state and topology (ARPANET, many others [2,5,8]).

3.1 Centralized Routing

Centralized routers [5] have the advantage that a complete, coherent "picture" of the entire network can be maintained at a single point. Decisions can be based on

the greatest possible amount of information, and attempt to optimize resource usage over a large area. However, centralization has several serious disadvantages. Communication overhead is involved in the collection of status reports from and the dissemination of routing decisions to the individual packet switches. The reliability of the communication paths to the central router (and that of the central site itself) is critical to the entire network.

The communication overhead with a centralized router more practical in a virtual circuit network, since routing is done only at circuit setup time. Such an approach is harder in datagram networks such as those based on ARPA IP because routing must be done on a per-packet basis (although one might "cache" routing information at the switches to minimize overhead).

3.2 Distributed Routing

An alternative is *distributed routing*, where each switch makes its own routing decisions based on a "local view" of the network. Switches usually exchange information with each other, either automatically or on a demand basis, thus maintaining a composite "snapshot" of the network. Such algorithms work well in datagram environments. They have other advantages, such as improved flexibility and reliability because of the lack of dependence on a central site.

For these reasons, distributed routing algorithms are popular, especially in datagram based networks such as the ARPA Internet, and I recommend its use in amateur packet radio. In the remainder of this paper, I will assume the use of distributed routing.

4. Routing Implications for Addressing

A packet switched network consists of a collection of nodes acting as packet sources, sinks and relay points. Depending on the topology of the network, a node may have to decide where to send a packet not destined to itself. In some cases, this is trivial. If the nodes share a transmission media allowing each node to communicate directly with all other nodes (e.g., Ethernet, closely spaced terrestrial packet radio nodes or nodes sharing a satellite channel) routing becomes trivial; packets can be send directly to the destination. Similarly, if the node is on a "stub" (i.e., it cm only communicate with one other mde) there is obviously only a single possible choice.

In the general case, however, packet network nodes are only partially interconnected with links and a packet must often be sent first to the neighbor which can best relay it onward to the destination.

One solution that works well in small networks (such as the ARPANET) is for each node to maintain a list of all other nodes in the network, giving the appropriate neighbor to which packets for each destination should be sent. (It is unspecified at the moment how these entries are determined.) As the network grows, however, each node's routing table will grow as well and the total amount of memory required at all nodes for routing tables will grow as the square of the number of nodes. Clearly, much memory could be saved if the list entries for nodes sharing the same "next hop" could somehow be condensed. This is possible if the addresses, instead of being arbitrary numbers, are related to the location of the node within the topology of the network.

At a node far removed from a given set of destinations, it is likely that the same neighbor would be used to reach any node within this set of destinations. If their addresses are "similar," in some sense, then it might be possible to "condense" these addresses into a single routing table entry.

5. Address Assignment Within IP

Bearing in mind the desirability of somehow encoding the topological location of a node into its address, I will now turn to the specific problem of address assignment within the ARPA Internet Protocol, IP.

An IP address field is 32 bits wide. IP addresses are further subdivided, primarily for administrative reasons, into three classes: A, B and C. The major difference between these three classes is the number of bits within this 32 bit field that may be assigned by the network administrator and how many are assigned by ARPA. This procedure is necessary if a network is ever to communicate with the existing ARPA Internet, since two sites might pick the same IP address unless there was some form of central coordination.

Thanks to the foresight of Hank Magnuski, KA6M, ARPA has assigned a Class A network number to amateur packet radio. This is a very valuable commodity, in that it fixes only the first byte of our addresses (to be decimal 44), leaving us the largest possible number of bits for our own use while keeping open the possibility of direct interconnection with the ARPA Internet. With the remaining 24 bits, we can address 16,777,216 nodes, easily enough to give every amateur in the world his or her own IP address if we allocate them efficiently.

Since AMPRNET is to be primarily a terrestrial radio network, it seems reasonable to encode a node's geographical location into its IP address. However, amateurs are distributed very unevenly throughout the world. Schemes that are based solely on geographic coordinates (e.g., grid squares), although aesthetically pleasing, are inefficient because they concentrate most of their address space over the poles, places with remarkably few amateurs.

Clearly a more efficient scheme is needed; one possibility is the binary tree. One way to illustrate this form of address assignment is with the game "Twenty Questions." Experienced players of this game know that the best strategy consists of asking questions for which "yes" and "no" answers are equally probable. In information theory, this corresponds to "maximizing the entropy of the source." For example, suppose that half of all the amateur packet stations in the world are in the United States. Then it would be reasonable to assign the first bit of the 24-bit address subfield to mean "US/non-US." Within the United States, one might determine that half of the packeteers are east of the Mississippi River and half are west, and so forth. Eventually, you reach a single "RF community" and you would assign the remaining address bits sequentially to the individual amateurs in that area.[1]

A major practical advantage of such a scheme is that the job of assigning addresses can be delegated to a hierarchy of organizations. An international organization (e.g., the IARU) would define only enough leading bits to uniquely designate each region or country in the world. National organizations within countries would then assign additional bits denoting regions within the country based on national concerns (i.e., the ARRL in the United States might handle this job based on American geopolitical boundaries). Other countries would have maximum freedom to devise their own national level addressing plans which might take into account unique national

1. This is Huffman encoding, similar in principle to the popular CP/M programs "SQUEEZE" and "UNSQUEEZE." Huffman coding compresses files by recoding them with variable length "characters" assigned according to the relative character distributions in the file.

requirements or conditions. At the lowest level, an individual packet station would only have to contact his local packet radio coordinating body for a specific address assignment, and these "front line" organizations would have maximum flexibility in devising an allocation scheme suitable for the local environment. Individual assignments would then be forwarded back up the organizational hierarchy (or maintained in a "well known" directory server) so that the network as a whole may have convenient access.

Since node addresses in a given area have common prefixes, it is likely that a distant node would only have to keep a single routing table entry for a large collection of nodes. For example, a packet switch in New York would only have to maintain the information that all packets to nodes west of the Mississippi are sent to node X, thereby "condensing" half of the packet nodes in the USA into a single routing table entry in the New York switch.

Depending on the network topology and address assignments, routing table entries may consist of variable length prefixes. These prefixes might vary from 0 bits long (corresponding to a "wild card" or "default" routing entry to be used when on the end of a stub, for example) to a full 32 bits when used to describe an special entry for a specific address. The latter case would be useful to handle special cases, such as point-to-point connections via satellite, or a node whose entry cannot be condensed with any other existing entry.

There is no guarantee that a routing table would not be larger than average if a node were located near a boundary in the address scheme, e.g., the US/Canadian border. However, appears that such a scheme would reduce the AVERAGE size of routing tables in the network. More work on this problem is needed, particularly a comparative estimate of the routing table sizes and growth rates for a variety of address boundaries and population distributions.

It should be noted here that the issue of hierarchical address assignment is drawing much interest in the ARPA community. Currently, IP addresses are assigned according to a two-level hierarchy: a Class A, B, or C "network number" part and a host part. Assumptions are made by the rest of the Internet that all hosts within a network (even a Class A network with 16 million hosts) are capable of "direct" connectivity without (externally visible) routing.[2] Several people have proposed that extra, optional levels be added to the two-level hierarchy, and four RFCs (ARPA memos) have been released with various proposals over the last several months. As of this writing, the issue is not yet settled.

6. Implementing a Distributed Routing Algorithm

A variety of distributed routing algorithms have been used, with the ARPA Internet serving as one important example. I will now describe an algorithm often used in Internet Protocol networks; many variations exist on this common theme.

For each destination, a routing table entry contains the following information:

1. The hardware interface on which such packets should be sent.
2. The node to which such packets should be addressed at the link level (same as the destination if the hardware can reach it directly, the link address of an intermediate gateway otherwise).
3. A *metric* that indicates the route's "cost". (Cost is typically just a "hop count," although it might also indicate the relative speed or loading factor of a link).

Each node starts with its routing table containing only its directly accessible neighbors ("neighbors" could mean a node on the far end of a point-to-point link, or, collectively, all the nodes on a shared network such as an Ethernet or local packet radio channel). The metric for each neighbor reflects the cost of the link to that neighbor.

Each node periodically broadcasts its routing tables to all neighbors. When a node receives such a table broadcast from a neighbor, it examines each entry to see if it refers to a destination that was previously unknown in its own routing table, or reflects a metric that is lower than the value associated with an destination already in the node's routing table. If either condition is true, then the node inserts the new entry into its own table after incrementing the metric to indicate the "cost" of the link to its neighbor.

In this way, connectivity information "diffuses" throughout the network, and packets are routed along paths that favor the minimum cost or hop count (depending on the meaning of the metric). When a node receives a routing table entry from a neighbor that contains a metric equal to or higher than an entry already in its own table for the same destination, the node might decide to accept the new entry anyway, keeping it in reserve when the preferred route fails.

To assure rapid recovery from a link failure or network reconfiguration, nodes often "poll" their neighbors periodically to assure themselves that they're still there. If a poll fails for some period of time, all routing table entries referring to that neighbor are removed, and an attempt is made to disseminate this information to the other neighbors that are still up.

As mentioned, many variations and enhancements are possible on this basic theme. For example, it has been observed that "good news" (the availability of a new node or link) "travels fast," while "bad news" (the failure of a node or link) "travels slowly." The polling rate is clearly a tradeoff; frequent polls minimize the time needed to detect and recover from a failure at the expense of extra network traffic. Other schemes attempt to avoid polling by acting only when a local client "complains" that a given node appears to be inaccessible.

With certain algorithms, it is possible to have transient "routing loops," where packets are forwarded endlessly. Fortunately, this need not be catastrophic in a network based in IP because the "time to live" (TTL) field bounds the number of hops a datagram is allowed to make. As long as the updated routing information is allowed to propagate, however, the network will eventually recover.

One problem that can occur in such a distributed scheme is that a node may advertise, either accidentally or maliciously, that it can reach *every* other node in the network with zero cost. Other nodes may then be gullible enough to accept this information and decide to route every packet to the offending node which discards them, effectively crashing the network [3]. It may be necessary to establish "sanity checks" or encryption-based procedures to establish the authenticity and reliability of routing information.

2. The ARPANET appears as a single Class A "local" network in the ARPA Internet, even though it spans the continental US and parts of Europe and the Pacific. Even though the ARPANET is not fully interconnected at the physical level, it does its own internal routing and thus appears as a fully interconnected network to the IP gateways.

7. Conclusions

I have only superficially scratched an involved topic, one that has been the subject of many books and learned journal articles. Nevertheless, I believe I can make several early recommendations that should ease the construction of our network and experimentation with practical routing algorithms:

- The use of a common datagram protocol at the network level (i.e., the ARPA Internet Protocol, IP) greatly simplifies the routing problem. Since a datagram network does not need nor guarantee absolute reliability, a wider variety of routing strategies may be considered. Routing protocols exploiting a datagram network's ability to efficiently broadcast routing information may also be used. Datagram networks can take full advantage of routing algorithms that dynamically balance link traffic. With IP, the sender always has the option of taking partial or full manual control of routing with the "source route" option, if desired.
- Network addresses should encode, in a hierarchical way that conserves address bits, the location of a node to reduce the amount of routing information that must be stored at each node and propagated throughout the network.
- A distributed routing algorithm should be used to avoid dependence on a central site and to allow maximum flexibility.
- Early emphasis should be made on establishing a standard protocol for the exchange of routing information (the ARPA Exterior Gateway Protocol, EGP, may be suitable for this purpose).
- Existing routing algorithms, particularly those used internally in the ARPANET and in the ARPA Internet should be investigated and tested to determine their suitability for widespread amateur use.

8. References

[1] Cerf and Kirstein, "Issues in Packet Network Interconnection," Proc. IEEE, November 1978 (special issue on Packet Communication Networks), p. 1386.

[2] L. Kleinrock, "Principles and Lessons in Packet Communications," p. 1320, same issue.

[3] R. Kahn et al, "Advances in Packet Radio Technology," p. 1468, same issue.

[4] L. W. Tymes, "Routing and Flow Control in TYMNET," IEEE Transactions on Communications, April 1981, page 392 (special issue on Congestion Control in Computer Networks).

[5] M. Gerla, "Controlling Routes, Traffic Rates and Buffer Allocation in Packet Networks," IEEE Communications Magazine, November 1984, p. 11 (special issue on Progress in Computer Communications).

[6] J. Hahn and D. Stolle, "Packet Radio Network Routing Algorithms: A Survey," p. 41, same issue.

[7] D. Clark, "Names, Addresses, Ports and Routes," ARPA RFC 814.

[8] W. Hsieh and I. Gitman, "Routing Strategies in Computer Networks," IEEE Computer, June 1984, p 46.

[9] J. Wescott, "Issues in Distributed Routing for Mobile Packet Radio Networks," IEEE

[10] D. Mills, "Exterior Gateway Protocol Formal Specification," ARPA RFC 904.

[11] R. Hinden, A. Sheltzer, "The DARPA Internet Gateway," ARPA RFC 823.

[12] P. Mockapetris, "Domain Names - Concepts and Facilities," ARPA RFC 882.

[13] P. Mockapetris, "Domain Names - Implementation Specification," ARPA RFC 883.

X.3 AND X.28 PROTOCOLS FOR TERMINAL NODE CONTROLLERS

Douglas Lockhart, VE7APU
9531 Odlin Road
Richmond, B.C. V6X 1E1
(604) 278-5601

Abstract

This paper proposes the adoption of an extended version of CCITT recommendations X.3 and X.28 for use in Amateur radio Terminal Node Controllers. The various X.3 parameters andX.28 commands and service signals (messages) are outlined and the extensions in place in the V-2 software implementation on the VADCG (Vancouver Amateur Digital Communication Group) TNC are discussed.

Terminology

This paper is semi-tutorial in nature and is intended for Amateur packet radio operators. Some of the terms used in the official X.28 protocol document have been translated into terms that hopefully may be more familiar to the readers. For example, the word 'TNC' is used instead of 'PAD' and the word 'terminal' is used instead of 'DTE' and in actual usage may be a microcomputer running host programs or a terminal emulation program. The terms I am using come from a variety of disciplines. My apologies if this makes it more difficult to understand rather than less so.

The Problem

In 1979, the VADCG TNC was the only one available and the software I had written for it only allowed two commands - control-Y caused a disconnect and a call sign followed by control-X caused a connection to be made. The very limited number of commands and the availability of only one version of software made it very easy for the new user to learn to use even though it was not very flexible.

The situation at the beginning of 1985 is quite different. There are now about 6 different TNCs available and three different protocols available, most with several different flavours or versions. Each combination of TNC and software has its own unique set of user commands. The more recent versions have a large and very flexible set of commands. In order to use the TNC efficiently, the user must learn his TNC's commands. With 40 or more commands frequently available, this becomes time-consuming. Furthermore, when the user tries to use the same commands on a different TNC or on the same TNC using a different link-level protocol, the commands are likely not to work and another set of commands must be learned. This learning process may be slowed by lack of advice from users with different command sets.

But perhaps the most serious problem that the lack of standardization of the command set causes is that application programs written for host computers will not be transportable. A special version for each case will have to be written. This situation has the potential of getting worse as new levels of protocols are developed, new TNCs become available and as new programmers begin to write software for TNCs.

The Solution

The commercial world ran into these same problems in interfacing ASCII terminals to the Packet Switched Public Data Networks (PSPDN) and developeda set of standards to solve their problems. The CCITT adopted Recommendations X.3 and X.28 in 1977. Recommendation X.28 defines the commands the terminal user sends to the PAD (Packet AssemblerDisassembler) and the service signals (messages) sent from the PAD to the terminal user. The PAD performs a similar function to the TNC (Terminal Node Controller) in the Amateur environment. Recommendation X.3 defines a set of parameters in the TNC which tailor its control of the terminal. In the ISO protocol model, X.3 and X.28 are a Level 6 or Presentation level protocol. To quote the OSI draft, "The purpose of the Presentation layer is to represent information to communicating application-entities in a way that preserves meaning while resolving syntax differences."

In August1984 Ireceivedinformation on the CCITT X.3, X.28 protocols and realized that these standards could solve most of these problems if they could be implemented on all TNCs. On August 20, 1984 I wrote to the ARRL Ad Hoc Committee on Amateur Radio Digital Communication proposing the adoption of these standards. I also began recoding the V-2 software I was writing to use these standards. The proposal received a favourable response at the September, 1984 Committee meeting. The implementation described in this paper was completed in October, 1984. At the Committee meeting in November, 1984 I was asked to prepare a paper detailing my recommendations for a protocol based on the CCITT X.3 and X.28 protocols to be used in the Amateur packet radio environment. This is that paper.

The environment that the X.3/X.28 protocols were intended to work in is very similar to that found in most Amateur packet radio stations. However there are some minor differences which necessitate some deviation from the official recommendation. X.3/X.28 was intended to interface ASCII terminals and because of this, it only expected 7-bit ASCII characters to be used. In the Amateur environment, it is sometimes useful to transmit 8-bit binary data such as in object code. Some modification and extension of these protocols has been made in this recommendation to accommodate this type of usage.

Additionally, the terminal user in the commercial environment had no control over the tailoring of the operation of the data link level software. This recommendation includes extensions to standardize some of this control the Amateur user is already familiar with. While an attempt has been made to keep the X.28 commands and messages as similar to the official recommendation as possible, several have had to be changed to be more meaningful in the Amateur packet radio environment. These changes are similar to the type of changes made by commercial users when adapting these protocols to their own environments.

Recommendation X.28

The first problem encountered in establishing communication between the terminal and TNC is to initialize the TNC to the proper data rate and data format oftheterminal. A sequence of characters are *sent* by the terminal to the TNC until the TNC recognizes the speed and format of the terminal and responds with an initial message. This function is commonly known as 'AutoBaud' or 'Autospeed' or 'Adaptive speed'. UnfortunatelyX.28 does not define any character sequence for implementing this function and in practice there are several differentmethods employed in the data communications industry. The current V-2 implementation on the VADCG TNC uses alternating '.' and ',' (period and comma) for detection of Baud rate and format. This system is superior to single character AutoBaud implementations because it can detect the type of parity and number of data bits used in addition to the Baud rate. In some applications it may be desireable to have the TNC always come up at a fixed Baud rate and data format and in this case the autoBaud function may be unnecessary.

After communication has been established between the terminal and TNC, both data for the link and X.28 commands for the TNC may be sent by the terminal user to the TNC. Likewise, messages generated by the TNC and data from the link can now be passedtothe terminal at this time. Since both data and commands are being multiplexed over the TNC/terminal connection, the question arises as to how to differentiate the two types of data. The TNC can operate in two modes: while in 'command mode* characters received from the terminal are interpreted as commands and while in 'data transfer' mode the characters are interpreted as data intended to be passed on the link. The TNC will go from data transfer mode to command mode when it receives the X.3 command escape character defined by X.3 parameter #1. This command escape character may be set to any value by issueing an X.28 command to modify the setting of an X.3 parameter.

The problem of transmitting the command escape character as data is solved by transmitting two escape characters in succession from the terminal. This sequence is recognized by the TNC and causes the TNC to pass a single escape character to the link. By using this 'byte stuffing' technique it is possible to transmit any combination of characters as data to the link. This technique is convenient for terminal users and for file transfer of ASCII files but it is not full data transparency. For transmission of binary files by host computers whose file transfer program cannot handle this 'byte stuffing' technique, another system must be used. X.3 parameters may be set so that the TNC does not recognize any characters as having special meaning - including the command escape character. (See X.3 parameters 3, 12, 13 and 15) Now we have full data transparency.

But we have a new problem. After we have transferred our transparent data, how do we get back into command mode? We can't use a command escape character to do it so it has to be done another way. The setting of X.3 parameter 7 to a value of 8 causes the TNC to go into command mode when a break signal is generated by the terminal. Some terminals and computers are not capable of generating a break signal but can get back into command mode by resetting the TNC causing it to return to the initial default values of the X.3 parameters.

Resetting of the TNC seems like an extreme measure and other alternatives for users incapable of generating a break signal should be found. This is an area where the standard may need to be extended to meet the special needs of the Amateur Radio environment. My suggestion is to use a special sequence of a number of characters followed by sufficient time to cause the idle timer to expire. (See X.3 parameter 4) At expiry of the timer, the TNC could check the preceding characters and enter the command mode if a match is found. This method should be virtually foolproof because a computer would never allow this much wait time while sending a transparent file. This is an area of further study.

Commands

X.28 commands are not case-sensitive. They can be typed in either upper or lower case or mixed case with the same results. Commands are terminated by carriage return <cr> or by'+'. Officially, X.28 has defined the following 8 commands:

SET <parameter list>
    To change X.3 parameter values
PAR? <parameter list>
    To display the current value of specified X.3 parameters.
SET? <parameter list>
    Sets and then displays the current value of specified X.3 parameter values
PROF <identifier>
    To set the X.3 parameters to a standard set of values
STAT - Request status of TNC
CLR - To terminate a link connection
RESET - To reset a link connection
INT - To transmit an interrupt packet

This is a fairly short list of commands. This is because the function of most of the TNC commands we are familiar with is done by modification of the X.3 parameters.

There are a number of problems with using this command set in the Amateur packet radio environment which I will discuss in the following paragraphs.

X.28 does not define a command used to establishaconnectionbecausethe format of such commands is very environment-specific. In the V-2 implementation the following command is used to establish a link-level connection:

CONNECT <call-sign><cr>

The call-sign is entered as an ASCII field in either upper or lower case. It is converted to upper-case and padded on the right with ASCII blanks. In the V-2 implementation this field is 7 characters long with the 7th character identifying an extension to the call sign. In implementations for other protocols, this field may be only 6 characters long. Additional information may need to be specified when establishing a connection when digipeating is used and when network level protocols are developed for Amateur packet radio. This information may optionally follow the call sign in the 'CONNECT' command but this is a subject for future consideration for incorporation into a standard. With a network level it may be desireable to use the 'CONNECT' command to establish a network level connection and use 'LINK' *to* establish a link level connection independently.

The X.28 'CLR' (clear) command is used to terminate a connection. Unfortunately, this command does not correspond to current Amateur packet radio conventions where the 'DISCONNECT' command is usually used. I propose we use the 'DISCONNECT' command for this purpose. When network level protocols are developed, the 'DISCONNECT' command can be used to terminate a network level connection and the 'UNLINK' command can be used to terminate a link level connection.

The X.28 'INT' command has no counterpart in the Amateur packet radio environment at the present time. There is no such thing as an 'interrupt

packet' in current link level protocols. When higher level protocols are implemented, there may be a need for this command. This is a subject for future consideration.

The 'SET' command should be implemented exactly as described in the X.28 recommendation. As an example, to set X.3 parameter 5 to 3 and parameter 21 to 128 type the following:

SET 5:3 21:128<cr>

The parameter numbers and values are entered as pairs of decimal numbers. Any blank or non-numeric character can be used as a separator between the numbers. The above could be entered as follows:

set 5 3 21 128<cr>

The X.28 'PAR?' command parameters is a list of X.3 parameter numbers whose values are to be displayed. The parameter numbers are entered in decimal.

The X.28 'SET?' parameters are the same as for the 'SET' command above. The function of the 'SET?' command is basically similar to that of the 'SET' command followed by the 'PAR?' command and because of this redundancy it has not been supported in the current V-2 implementation. However, since it may be more convenient, this command should be considered optional.

The X.28 'PROF' command sets the X.3 parameters to a particular set of values or 'profile.' The identifier in the command is a decimal number which identifies the particular profile desired. X.28 defines two profiles - simple and transparent but many networks offer other profiles tailored to the characteristics of a particular terminal or application.

The transparent profile is used when the TNC needs to be "invisible" to the user. Commands can not be used, no messages are generated by the TNC and editing and echo are not supported.

The simple profile supports commands and uses a set of X.3 function which require a minimum of terminal features. Messages are generated by the TNC and echo and flow control are enabled. Data is forwarded whenever a control character is received by the TNC.

The 'initial profile' is the set of X.3 parameters that are in effect when the TNC is initialized or powered up. It is usually similar to the simple profile. TNCs that support non-volatile RAM (NOVRAM) can have the ability to tailor their initial profile while others can only support a fixed initial profile burned into EPROM. The 'PROF' command is particularly useful in the latter case because it can usually reduce the amount of time required to tailor the X.3 parameters after powering on the TNC. It can also be useful when the TNC is sometimes switched to a different function where substantially different characteristics are required. This paper recommends that the 'PROF' command should be considered optional but desireable. It is not supported in the current implementation of V-2 but will be implemented in a future release.

The 'STAT' command takes no parameters but causes the TNC to display the current connect status of the TNC. This command is not presently implemented on V-2 since the connect status of the TNC is indicated by two other methods. However, this command should be implemented since TNC software capable of multiple links and connections will. become available and the connection status of a TNC may be difficult to determine without this command.

The 'RESET' command currently is not supported because the current link level protocols do not support a reset packet. This command should be implemented when new protocols are developed which support this function.

You will note the very small number of commands needed by X.28 to control a TNC. Most functions are performed by the setting of X.3 parameters using the 'SET' command. The reduction in the number of messages and commands can reduce the learning period for a non-English speaking user and possibly reduce the time for programmers to recode the software for use with a different language.

In general, functions which control an on/off type of condition or control a value that can be specified in a single byte can be controlled by X.3 parameters. For example, the TNC either echoes the characters from the terminal or it doesn't. Another example is the setting of the number of retries attempted on the link.

On the other hand, functions which cause immediate action or require the entry of multiple characters are unsuitable for incorporation into X.3 parameters and so require specific commands. An example ofthis is the 'CONNECT command which requires the provision of a call sign and causes an immediate change in the operation of the TNC.

One function common to all TNC command sets is the ability to set the station call sign and so should beincorporatedinto a standard. I recommend the following command for this function:

MYCALL <call-sign>

This format is similar to that of the 'CONNECT' command described above. At the present time the V-2 implementation uses the 'CALLSIGN' command to perform this function but the 'MYCALL' command mnemonic is more familiar to users of TNCs and so I will be changing it in future releases.

The commands described in this section are not meant to be the only commands that should belong in a TNC. They are intended to be a set of commands that should be standardized in every TNC because of a common need for these functions. If a specific function is only used in one implementation there is no need for standardization. However, if a function is common to more than one implementation, then there should be an attempt made at providing a standard way to controlthatfunction. No doubt, as Amateur packet radio develops, new commands will have to be incorporated into this recommendation.

To summarize this section, this paper recommends the incorporation of the following six commands into every TNC.
SET, PAR?, CONNECT, DISCONNECT, MYCALL, STAT.
The PROF and SET? commands are optional.

X.28 Messages

X.28 messages are frequently called service signals. The public packet-switched networks do not usually implement the official X.28 message set precisely as it is defined. This is because the messages are not human engineered and in many cases may not be suitable for a specific environment. This is also the case in the Amateur Radio environment. However a number of the messages can be used. The following list is a subset of the official X.28 message set that I am recommending be used exactly as defined.

Linefeed
Acknowledgment of a command.

*
Command mode prompt signal.

XXX
Indicates that the line delete function is completed.

ERROR
Command error.

PAR <n:n>
Response to SET and SET? commands. The n's indicate the parameter number and value in decimal respectively.

PAR <n:INV>
Response to an invalid parameter setting in a SET or SET? command. n is the parameter number.

The following list contains messages which are not exactly as defined by the official X.28 recommendation but which I recommend be used in the Amateur packet radio environment.

LINKED <call-sign>
Indicates that a data link connection has just become establishedor aresponse to the STAT command when a data link connection is established.

UNLINKED
Response to the STAT command when a data link connection is not established.

LINKING <call-sign>
Response to the STAT command when a data link connection is being attempted.

UNLINKING <call-sign>
Response to the STAT command when a data link connection is being broken.

UNLINKED <call-sign> <reason>
Indication that a data-link connection has been broken or could not be established. The reason is given after the call-sign and can be one of the following:

BUSY - The indicated station is busy.
REM - The remote data link partner terminated the link.
REMERR - The remote station detected a protocol error.
NORM - Local DISCONNECT command.
NORESP - The station did not respond.
PROT - Protocol mismatch.
ERROR - Terminated by link protocol error.
TIMOUT - Excessive timeouts on link.
INV - Requested facility not supported.

Some of the reasons refer to facilities not supported in all link-level protocols. In this case, these reasons need not be used. However, when these facilities are available, they should be implemented as shown. Other reasons for link termination will be the subject of future standardization attempts.

It should be noted that the words, "link, linked, unlinked, etc." are used instead of the words, "connect, connected, disconnected, etc." even though the latter terms are in more common use at the present time. This was done to avoid future confusion because of terminology. In the IS0 model, "connections" may be established at all levels. There are link level connections, network level connections, transport level connections, session level connections, etc. In the future, TNCs will likely have software to support multiple protocol levels, not just the link level as at present. TNCs capable of supporting multiple network level connections and multiple link level connections simultaneously will likely appear. When this happens, the term "connection" used without qualifiers will be ambiguous and cause some confusion. The term, "link" refers only to the data link layer and its use should reduce this potential problem. I believe that the term, "connection" without qualifiers should only be used to refer to a network level connection.

X.28 does not define the initialization message which apears when the TNC is initialized and I am making no recommendation in this area.

Recommendation X.3

In 1984, the official X.3 recommendation defined22 parameters used to tailor the way the TNC controls the interface between the TNC and terminal or host computer. Unlike the X.28 recommendation which needed substantial modification to adapt it to the Amateur packet radio environment, these X.3 parameters can be implemented almost exactly as describedin the official recommendation. For the Amateur packet radio environment, this paper recommends some extensions to this set to provide additional tailoring of the terminal interface, tailoring of the operation of the link interface and to control operation of the network level if implemented.

In October of 1984, the V-2 implementation on the VADCG TNC supported an additional 14 parameters beyond the 22 used by the official recommendation to provide this additional control. The parameters supported in this implementation are summarized in Table 1. The officialX.3 parameter values which are not supported are indicated and extensions to the standard are also indicated. All parameters beyond 22 are extensions. The default values in the distributed software are also indicated. These defaults should not be considered as part of this recommendation but they do serve as an example of an 'initial standard profile'.

In this implementation the X.3 parameters have been set up as a contiguous string of bytes in memory. The first byte being the value of parameter 1 and it is recommended that the parameters be implemented in this way. Each byte has a set of legal values defined by this recommendation. The software should not allow non-legal values to be set into these parameters.

Parameter 1 defines the character sent to switch the TNC from data transfer mode into command mode. While the official standard only permitted values of 0 and 32-326 to be specified, I have extended this range to allow all 8-bit byte values. This was done because TNCs are frequently used with host computers rather than ASCII terminals and so are capable of transmitting80bit characters. A value of 0 set in parameter 1 prevents escape from data transfer mode. by the transmission of a command escape character. Only one command can be issued from the terminal per escape. The TNC reverts back to command mode after each command is issued. Reception of the escape character also caused any accumulated data received from the terminal to be forwarded as a packet.

Parameter 2 controls whether the TNC will echo characters received from the terminal.

Parameter 3 defines characters which the TNC recognizes as a signal to forward any accumulated data received from the terminal and transmit it as a packet on the link. See parameters 1, 4 and 33 for other conditions which cause accumulated data to be forwarded. Note that values in parameter 3 can be added together to get some forwarding combinations that are not standard in the official recommendation. For example, a value of 10 can be specifiedto cause forwarding on carriage return (CR), DEL, CAN and DC2.

Parameter 4 defines a time interval of terminal inactivity which is used to forward any accumulated data as a packet. The timer is started each time a character is received by the TNC. If it expires, the data is forwarded. This forwarding method is disabled if the parameter is set to 0.

Parameter 5 defines the type of flow control indication is to be used by the TNC to regulate the flow of data from the terminal. In addition to the X-ON/X-OFF flow control method defined in the

official X.3 recommendation, the TNC can also control the RTS (Request to Send) line on the terminal interface to indicate whether it is ready to accept data or not. An active RTS line indicates that the TNC is ready to accept data. This extension was necessary to support the transfer of non-ASCII data (such as object code) by host computers to the TNC while still retaining flow control capability. The X-ON/X-OFF protocol is not capable of doing this.

Parameter 6 controls the use of X.28 messages by the TNC. These messages can be disabled as is frequently necessary when the TNC is connected to a host computer rather than a terminal or when the TNC is to remain transparent to the user. The network dependent format service signals are not supported because this requires a protocol layer that is not used in Amateur TNCs at the present time. It is a subject of future consideration when the protocols are in place that can support this function.

Parameter 7 determines the action of the TNC when it receives a break signal from the terminal. The only values that are supported by the V-2 implementation are 0 and 8. Value 8 provides another way to escape to command mode for terminals which support the break signal. It may be particularly useful for host computers wishing to switch from a transparent mode of operation after transferring a file. Values 1,2,4,5, and 21 are not supported because there are no interrupt packets, reset packets or any way to indicate the detection of a break in a packet using the current Amateur link level protocols. Their use should be reconsidered when higher level protocols which support these functions become available. Although not implemented at present, value 16 which supports the ability to flush TNC data for the terminal is useful.

Parameter 8 can be set so that the TNC will discard all data for the terminal. The only use for this feature at the present time is for testing purposes. When session level protocols are implemented in Amateur packet networks, this feature will become useful. It is easy to implement.

Parameter 9 controls the number of padding characters (ASCII nulls) which are sent to the terminal after every carriage return character. This is frequently required by hard copy terminals which require extra time to perform a carriage return. Some video terminals and terminal emulation programs require extra time as well. In the V-2 implementation the padding characters are actually sent but this function may be implemented by adding the equivalent amount of time fill as well. The official X.3 recommendation only allowed up to 7 padding characters but the V-2 implementation allows up to 255paddingcharacters or equivalent time fill.

Parameter 10 controls the point where long lines are broken or folded so that they display as multiple lines. This has its main use with RTTY terminals to prevent the last characters in a long line from overprinting at the end of the line. Some video terminals do not support line folding and may need to use this feature.

Parameter 11 indicates the speed of the terminal. Take special note that this parameter cannot be changed by the X.28 SET or SET? commands. It is a read only parameter that is setbythe TNC initialization process or by the AutoBaud routine. Its only use at present is to determine what speed the terminal is running at if this is not known. It will have other uses when session level protocols are used in Amateur packet networks. The values indicated as not being supported are Baud rates that the V-2 AutoBaud routine does not support. Baud rates not in the list which are implemented should use values above 18 and willbethe subject of future standardization efforts.

Parameter 12 defines the type of flow control signals which the TNC will act upon to control the flow of data from the TNC to the terminal. The functions provided by this parameter have been extended beyond the official recommendation in order to be able to pass binary data to the terminal or host computer. An additional flow control method using the CTS (Clear to send) line from the terminal has been provided. If value 2 is set, the TNC will only send data to the terminal when the CTS line is active. Note that X-ON/X-OFF flow control cannot be used while transferring binary data. When X-ON/X-OFF flow control is specified here, the X-ON and X-OFF (DC1 and DC3) characters are not forwarded.

Parameter 13 controls the insertion of line feeds by the TNC after carriage returns either going to or coming from the terminal and after carriage returns being echoed back to the terminal. Any combination of these insertions may be selected.

Parameter 14 controls the number of padding characters (ASCII nulls) to be inserted after every line feed being sent to the terminal. This feature is used with terminals which take a long time to do a line feed operation. This is somewhat similar to parameter 9 and equivalent time fill can be used instead of sending padding characters.

Parameter 15 controls whether editing operations can be performed on dataaccumulatedbythe TNC in data transfer mode but which have not been forwarded yet. Editing is always available in command mode. The characters acted upon for the editing functions are defined in parameters 16, 17 and 18. Editing is normally used with terminals rather than host computers. The V-2 implementation does not turn off idle time forwarding automatically when editing is specified as per the official X-3 recommendation. Care should be taken with the specification of parameter 3 to ensure that data is not forwarded before there is a chance to edit it.

Parameter 16 defines the editing character the TNC acts upon to delete the previous entered character from the unforwarded accumulated data. The V-2 implementation uses the range O-255 rather than 0-127 which is the official X.3 recommendation.

Parameter 17 defines the editing character the TNC acts upon to delete the unforwarded accumulated data. The V-2 implementation uses the range O-255 rather than 0-127 which is the official X.3 recommendation.

Parameter 18 defines the editing character the TNC acts upon to redisplay the unforwarded accumulated data. This is mainly used to redisplay edited lines on printing terminals which do not have the ability to erase characters. The V-2 implementation uses the range O-255 rather than 0-127.

Parameter 19 selects the type of signals that will be returned to the terminal when handling editing characters. The V-2 implementation allows for tailoring these signals for either printing or video display terminals. For example: for the delete previous character operation the TNC will return the character "/" for a printing terminal or the character sequence "<backspace>, <space>, <backspace>" for a video display terminal.

Parameter 20 is used to specify which characters will not be echoed to the terminal when echo is enabled by parameter 2. Note that the flow control characters X-OFF and X-ON are never echoed. Note that the values in this parameter can be added

together to get character sets which are not in the official recommendation. For example, a value of 6 may be specified which prevents echo of LF, VT, HT and FF.

Parameter 21 specifies whether the TNC will generate the parity bit in the data sent to the terminal and whether parity will be checked on data from the terminal. This parameter is not used in the V-2 implementation. The AutoBaud routine in V-2 generates a like parity to that supplied by the terminal but no action is taken on parity errors in data received from the terminal. The official X.3 specification seems to be vague as to what action is to be taken on parity errors or what type of parity (even, odd, mark, space, etc.) will be set by this parameter. However, I recommend implementation of this parameter when its function is clear and useful.

Parameter 22 specifies the number of line feed characters that the TNC will sendtotheterminal before stopping the sending of additional characters. The sending will recommence when an X-ON character is received from the terminal. This function allowstheterminaluser to read all the information before it is scrolled off the screen. The X.28 message "PAGE" is senttotheterminal to indicate the reason the data transfer to the terminal has been stopped.

Parameter 23 defines the number of additional characters that the TNC can receive from the terminal when the flow control method specified by parameter 5 acts to stop data from coming in from the terminal. This parameter and all remaining parameters described here are extensions to the official X.3 recommendation. This parameter was added to support a number of host microcomputers which do not act immediately on flow control signals. Some microcomputers only check flow control signals after the transmission of each line which could result in lost data if extra buffering capability was not available at the time the TNC acted to suspend the data flow.

Parameters 24, 25, 26, 27, 28 and 29 are extensions to allow for tailoring of the operation of the data link on the TNC.

Parameter 24 defines the number of consecutive link timeouts (no response that will be tolerated before "giving up". This value is used only when a 'CONNECT' or 'DISCONNECT' command is in progress. When a link is established, the value in parameter 25 is used.

Parameter 25 defines the maximum number of consecutive link timeouts that will be tolerated on a data-link connection before "giving up". This value is only used when the link is established. See parameter 24 for the value used when the link is being established or terminated.

Parameter 26 defines the amount of time the TNC will wait for a response from a data link partner. If this time expires before receiving the expected response the TNC takes remedial action which is usually a transmission polling the partner for another response. The V-2 implementation on the VADCG TNC suspends this timer when the link is busy (a signal is detectedonthe channel) so that timeouts are not caused by use of the communications channel by other stations.

Parameter 27 allows for improved usage of the TNC in situations where it is not possible to reliably determine whether the link channel is occupied or not. This is frequently the case with noisy links such as are encountered with satellite and HF channels. When a value of 1 is specified, the TNC ignores the status of the carrier detect line and always assumes that the channel is clear.

Parameter 28 defines the amount of time the RTS line will be active before the data is transmitted on the data lines. This is used to support transmitters and modems which are slow in turning on. This time delay is also used to allow sufficienttime for receivers to unsquelch so that the first part of the data frame is received without error. Note that the data will not be transmitted until the CTS line is active too.

Parameter 29 can be used to prevent links by other stations from being established. When the value is set to 1, the V-2 protocol responds to attempts to establish a data-link connection by indicating a 'busy' condition.

Parameter 30 is intended for future link control functions.

Parameter 31 controls whether the TNC will use the AutoBaud feature when re-initialized or use preset values for speed and format of the terminal interface. In TNCs which do not support non-volatile RAM, the value set by the terminal user will only be effective until the TNC is powered off.

Parameter 32 controls which packets received by the TNC will be passedtothe terminal. If set to 0, the TNC acts as a filter which prevents data in packets not intended for this station from being passed to the terminal. Other values to provide different filtering actions should be the subject of future standardisation efforts.

Parameter 33 specifies the maximum number of characters received from the terminal which can be accumulated before forwarding the data in a packet. This determines the maximum number of bytes in the data field of a packet. The actual maximum frame length transmitted must include the overhead of the various protocol layers in use as well as this value and the maximum value that can be specified may vary with different protocol implementations. Care should be taken when specifying this parameter so that frames which are greater than the link partner's ability to handle are not generated. I recommend that all implementations be capable of handling values of at least 128 characters.

Parameters 34through 38 are intended to be used to tailor the operation of the network layer. It is too early in the development of Amateur packet radio protocols to standardize network layer functions. The current use of parameter 34 in the V-2 implementation is shown in Table 1 but it is not the intent of this paper to recommend standards in this area.

Parameter 39 determines how the TNC controls the Receive Line Signal Detect (Carrier Detect) line on the terminal interface which is pin 8 on the standard EIA RS-232 interface connector. If set to 0, this line is always active. If set to 1, this line is only active when a data-link connection has been established. This feature is useful when the TNC is connected to a host computer.

Parameter 40 controls the masking of data bits in the characters being sent to and received from the terminal. If this value is 255, the TNC will pass 8-bit characters to and from the terminal. If the value is 327, the high order bit will be masked off (set to 0). Any bits can be masked off. The masking values are easier to understand when converted to binary.

Recommendation X.29

Although not discussed previously, it seems appropriate that CCITT recommendation X.29 be mentioned because it is closely related to the X.3 protocol. X.29 is a protocol which allows a remote

TNC to both inspect and change the X.3 parameter values. This allows an application program to coordinate its operation with that of the remote terminal. For example, the application program could disable echo by altering X.3 parameter 2 while a password was being entered. In the ISO model, X.29 would be called a Level 5 or Session level protocol because it provides the means to coordinate the operation of the remote Presentation level (X.3/X.28). A number of significant advantages of the X.3/X.28 protocols will not be realized until a Session level of some kind such as X.29 is implemented in Amateur packet networks but it is not within the scope of this paper to make recommendations on Session level protocols for Amateur use.

Implementations

The only full implementation of the X.3 and X.28 protocols on Amateur TNCs that I am aware of is with the V-2 protocol on the VADCG TNC or Ashby boards. The Ashby board requires a hardware modification for the software to work. I understand that Terry Fox, WB4JFI with AMRAD is working on implementing this protocol with AX.25 on the VADCG TNC with his daughter boardmodification. He has been sent a copy of this implementation. TAPR representatives have indicated that they will provide X.3/X.28 as an optional command set on the TAPR board at some future date. Phil Karn, KA9Q, who is developing packet software for the Xerox 820 board is also interested in using these protocols. From discussions I have had with representatives of many of the Amateur packet radio organizations, I feel there is widespread support for the development of standards based on the X.3 and X.28 protocols.

Coordination of implementation

Some problems may arise when implementing these protocols in other TNCs or with other link-level protocols. The following paragraphs are guidelines for implementors.

Not all TNCs will support all the functions defined in the X.3 parameters. An invalid parameter message should be returned when an attempt is made to set an unsupported value.

Functions which can be implemented in X.3 parameters but which are not defined in the current recommendation should be implemented in parameters above 40.

When these new parameters are used in an implementation an attempt should be made to coordinate these parameters with other implementations so that the same parameter is not used for different functions or that different parameters are used for the same function. On request, I have volunteered to act as a coordinator for this effort within the ARRL Ad Hoc Committee on Amateur Radio Digital Communication. The Committee will use its facilities to act as a clearing house for information on X.3/X.28 implementations for individuals who are planning extensions to these recommendations. Implementors should write to the Committee or directly to me at the address at the beginning of this paper.

Many functions are not suitable for control by X.3 parameters and will require extensions of the X.28 command and message set. An attempt to coordinate these extensions should be made. Implementors should communicate details of these extensions to me or the Committee as described in the previous paragraph.

Summary

The protocol described in this paper has been tested by many Amateurs in a number of countries with good results and it is estimated that several hundred thousand users of public data networks use the X.3 and X.28 protocols daily. The temporary disruption caused during the conversion from the old command set to the X.3/X.28 protocols is only a small inconvenience when compared to the advantages of having a standard set of commands and messages for all TNCs. This inconvenience becomes even less significant when one considers the potential advantages of the X.3 parameters when session level protocols such as X.29 are developed for Amateur packet networks.

The initial recommendations made here will likely be fine-tuned as a result of future discussions, feedback and coordination efforts but the author hopes that this paper will lead to the development of a widespread standard set of commands and responses for terminal users of Amateur packet radio equipment and to give programmers a standard interface to TNCs so that they may write applications that will work on all Amateur TNCs. The recommendations made in this paper do not restrict the commands and messages used by TNCs but do propose standard ways of controlling certain common functions. No doubt, as Amateur packet radio networks mature, other functions will need to be standardized and other presentation level protocols will be developed for special environments.

Reference

[1] Shanahan, Teresa A., "Packet Assembly/Dissassembly (PAD) CCITT Recommendations X.3, X.28 and X.29" Data Transfer newsletter, Transmission #9, April 1984, Omnicom Information Service.

TABLE 1 - X.3 PARAMETER VALUES

* = default value
\# = not supported
% = additional to X.3 standard

REFERENCE		
1 Command escape	0	not possible
	*27	ESC character
	1-26, 28-255	optional values
2 Echo	0	no echo
	*1	echo
3 Data forwarding	0	full packet only
	1	alphanumerics
	*2	carriage return
	4	ESC, BEL, ENQ, ACK
	6	carriage return, ESC, BEL, ENQ, ACK
	8	DEL, CAN, DC2
	16	ETX, EOT
	18	carriage return, EOT, ETX
	32	HT, LF, VT, FF
	126	all other characters below 32
4 Idle timer delay	0	no timer
	*32	default value
	1-31, 33-255	delay values in approximately tenths of a second.
5 Flow control to TNC	0	no flow control
	1	X-ON/X-OFF (data transfer)
	2	X-ON/X-OFF (data transfer and command)
	%*4	flow control with RTS line
6 Control of TNC service sigs	0	no service signals
	1	transmit service signals
	*5	transmit service and prompt sigs
	#8-15	network dependent format service sigs
7 operation on break	*0	no action
	#1	interrupt packet
	#2	reset packet
	#4	indication of break service signal
	#5	interrupt and indication of break
	8	escape from data transfer state
	#16	discard output to terminal
	#21	1 + 4 + 16 combined
8 discard output	*0	normal data delivery
	1	discard output to terminal
9 Carriage return padding	*0	no padding
	1-255	number of nulls inserted after CR
10 Line folding	*0	no line folding
	1-255	number of characters per line

11 Binary speed	0	110 bit/s
	1	134.5 bit/s
	2	300 bit/s
	3	1200 bit/s
	4	600 bit/s
	5	75 bit/s
	6	150 bit/s
		1800 bit/s
	8	200 bit/s
	9	100 bit/s
	10	50 bit/s
	#11	75/1200 bit/s
	12	2400 bit/s
	13	4800 bit/s
	14	9600 bit/s
	15	19200 bit/s
	#16	48000 bit/s
	#17	56000 bit/s
	#18	64000 bit/s
12 Flow control to terminal	*0	no flow control
	1	X-ON/X-OFF flow control
	%2	flow control using CTS line
13 Line feed insertion	0	None
	1	after carriage return to terminal
	2	after carriage return from terminal
	4	after echoed carriage return
	5	values 1 + 4
	"6	values 2 + 4
		values 1 + 2 + 4 (data transfer only)
14 Line feed padding	*0	none
	l-255	number of nulls after line feed
15 Editing	0	off
	*1	on
16 Character delete	*8	BS (backspace+ character (control-H)
	o-7, 9-127	other characters
17 Line delete	*24	CAN character (control-X)
	O-23, 25-127	other characters
18 Line redisplay	*18	DC2 character (control-R)
	o-17, 19-127	other characters
19 Editing service signals	0	no editing service signals
	1	editing for printing terminals
	*2	editing for display terminals
	#8	editing using characters from range 32-126
20 Echo mask	*0	all characters echoed
		no echo of carriage return
	2	no echo of LF
	4	no echo of VT, HT, FF
	8	no echo of BEL, BS
	16	no echo of ESC, ENQ
	32	no echo of ACK, NAK, STX, SOH, EOT,ETB, ETX
	64	no echo of editing characters
	128	no echo of all characters in columns 1 & 2 plus DEL
#21 Parity treatment	#0	no parity detection or generation
	#1	parity checking
	#2	parity generation
	#3	value 1 + 2
22 Page wait	*0	no page wait
	l-255	number of line feed characters before waiting

%23 Buffer cushion	*80	number of characters in cushion
	2-79, 81-254	other values
%24 Linked timeouts	*16	maximum consecutive timeouts linked
	1-15, 17-255	other values
%25 Unlinked timeouts	*5	maximum consecutive timeouts when not linked
	1-4, 6-255	other values
%26 Line timeout	*10	approximately 5 seconds
	1-9, 11,255	other values
%27 Duplex line control	*0	half duplex line control
	1	always assume that channel is clear
	#2	full duplex
%28 Clear to send delay	*32	approximately 1 second
	l-31, 33-255	other values
%29 Link control	*0	normal
	1	links to other nodes not permitted
%30 Unused link control parameter		
%31 Autobaud control	0	use fixed UART defaults
	*1	AutoBaud
%32 Received packet forwarding	0	only pass packets when linked
	*1	pass unlinked packets
%33 Maximum packet length	*200	maximum output packet data bytes
	3-199, 201-250	other values
%34 Network header 2nd byte	*0	NH second byte is 00
	l-255	other values
%35 Unused network control parameter		
%36 Unused network control parameter		
%37 Unused network control parameter		
%38 Unused network control parameter		
%39 RLSD (CD) line control	*0	always on
	1	indicates if link is established
%40 Data mask	*127	mask off high order bit
	255	pass all 8 bits in each byte
	0-255	range permitted

Formal Definition Meeting for the Packet Radio Experiment RUDAK to be included in AMSAT P3-C

Dr. Karl Meinzer DJ4ZC
Hans Peter Kuhlen DK1YQ

During the weekend February 15 thru 17, AMSAT-DL hosted a formal meeting to define the Packet payload in P3-C. The experiment has been named "RUDAK" for "Regenerativer Umsetzer für Digitale Amateur-Kommunikation".

Attending were:

Hans Peter Kuhlen DK1YQ
Program Manager RUDAK Project

Peter Gülzow DB2OS
Project Manager RUDAK project

Heinz Mölleken DL3AH
Ground Systems Manager RUDAK project

Werner Haas DJ5KQ
Vice President AMSAT-DL e. V.

Karl Meinzer DJ4ZC
President AMSAT-DL e. V.

A. General

After a brief review of the performance and capabilities of existing packet systems, the board set the objectives for the RUDAK payload as follows:

1. Compatability of the system with the present AX.25 standard and the existing Packet Radio boards (e. G. TAPR).
2. Regular Amateur communication equipment should be used without the need of modification or intrusion.
3. Moderate to small antennas should be sufficient for low bit error rates.

Relating to point 3., the board agreed on the nominal amateur station performance parameters as detailed in Annex A. With these in mind, the second point was analysed in great deteil with particular reference to link-performance and modulation techniques available with these results:

a. Link budget considerations require efficient techniques for the downlink which today can only be achieved using (transparent) SSB-equipment with demodulation at baseband (audio). This limits the practical achievable datarate to 1200 bits/s (RSM) or lower (BPSK); a performance better than 12 dB Eb/No can be expected.

b. The uplink could employ standard FM-equipment for straight FSK-modulation. Experiments by DB2OS showed that 2400 bits/s (biphase) can be handled by standard equipment without problems. It remeins to be investigated if 4800 bits/s (NRZ) also can be handled or if special measures are necessary to eliminate the influence of the DC-component (e. G. scrambling for spectrum shaping). Higher data rates cannot be achieved with standard radios. Technical papers reviewed indicate that with a discriminator type of demodulator 17 dB Eb/No are necessary for 2400 bits/s and about 15 dB for 4800 bits/s (FM-threshold), The meeting concluded that also BPSK for the uplink is

viable without intrusion inti equipment by using a high-power passive **BPSK**-modulator between transmitter and antenna or between exciter and PA. This approach imposes no restrictions on the data rate and yields also a better than **12 dB Eb/No** performance. The resulting spectrum needs to be investigated and bandwidth-limiting measures may turn out to be necessary. The board concluded **theat** in view of the long visibility of the satellite no **sgnificant** on board storage would be employed. The **uplink** using essentially ALOHA **signal**ling should have about six times the capacity of the downlink. On board storage should be sufficient to buffer about ten times the packet differential between **downlink** and **uplink (6-7 kByte).**

c. presently there is no suitable **ISO**-layer 3 network definition available, thus the payload initially should emulate the existing digipeater function as defined in the AX.25 version **2.0/Oct.** 84. If a more sophisticated level 3 protocol1 becomes available, the S/C will be updated accordingly.

B. Design decisions taken by the board

a. The board agreed on the following main features as design guidelines for the RUDAK experiment:
- Nominal **amateur** equipment as defined by Annex A required the selection of the following data-rates and modulation techniques:

Uplink: (24 cm)	2400 bit/s differential biphase PSK (+-90 deg) spectrum shaping TBD
Downlink: (70 cm)	400 bit/s differential biphase PSK (+-90 deg) spectrum shaping as used in AO-10

- continuous operation of the beacon in Mode L **(24/70** cm: independant **from** the transponder **passband** (and AGC)
- "Bulletin board" i. e. cyclic repetition of information packets containing
  .updated satellite status (telemetry)
  .orbit information (Keppler data) and present position (MA)
  .uplink parameter set to be used by Packet Radio Stations wishing access to RUDAK (to eliminate unnecessary trial and error experimentation).
  .etc.
- RUDAK programmes will be resident entirely in RAM facilitating software updates to be executed by **AMSAT** control stations via the regular **P3-C** command system.
- Packet first-in-first-out (FIFO) buffer **(6-7 kByte)** plus additional storage consistent with available memory to be used.
- continuous self-test of the s/w with error correction in case of soft errors and auto-recovery in case of problems.

b. The original RUDAK design constraints (power 5 W, Volume 5 litres, mass 5 kg) were reviewed. It was concluded that one large **P3-module (300x200x40** mm) would be sufficient to house the digital part of the experiment. The board was made aware that for a continuous operation of the RUDAK computer a considerably lower power consumption than 5 W would be desirable. If this turns out to be impractical, the availability of a stand-by mode with memory retention should be investigated. The transmitter and receiver of RUDAK will be built and integrated into the L-transponder by the group building the transponder.

C. A work assignment and schedule has been agreed upon consistent with the **AMSAT-P3-C** launch (Annex **B).**

d. The board elected H. Kuhlen, DK1YQ, to compile the full RUDAK specification for definition of hardware and software requirements including the interfaces to the"Integrated Housekeeping Unit" (IHU) and the Mode-L-transponder.

e. Offers of participation to interested AMSAT groups will be released after availability of the full specification set.

f. Development of a compatible ground MODEM ans its early publication will be initiated in parallel with the space segment development.

ANNEX A. (Link assumptions and calculations)

Both Mode-B and Mode-L link-scenarios have been investigated. Mode-B finally was rejected because the expected downlink-performance in the 2m-Band was considered unsatisfactory in Japan and European metropolitan areas. Also the lack of suitable spectrum space in the 2m-Band, the bulk and cost of the required 2m-antenna and the fact, that the U-transponder exists already, entered into the decision. For the sake of completeness, the links are also presented for Mode-B.

Ground station assumptions:

Mode B:	Receiving (2m)	Gant:	+9 dBi
		Tn:	1000 k
	Transmitting (70cm)	Gant:	+10 dBi
		P-Tx:	5+ w --27 dBWi
Mode L:	Receiving (70 cm)	Gant:	+10 dBi
		Tn:	1000 k
	Transmitting (24cm)	Gant:	+15 dBi
		P-Tx:	12 w -- 26 dBWi

All links are to be designed with 7 dB margin to cover the less than perfect equipment to be expected in the amateur-environment.

Mode-B links: (for reference only)

Downlink	P-Tx	5 dBW (3W)
	Gant S/C	3 dBi (min during spin)
	link at apogee	-168 dB
	misc losses in link and S/C	-3 dB
	margin	-7 dB
	Gant-ground	+9 dBi
	Received power ground Rx	-161 dBW
	Pn (400bit/s)	-173 dBW
	--- Eb/No	12 dB

Mode-B uplink (Assuming 2400 Bit/s FSK)

Gant ground	10 dBi
P-Tx (50 W)	17 dBW
link	-177 dB
misc. losses	-3 dB
margin	-7 dB
G-ant S/C	+9 dBi
Received power at S/C	-151 dBW
Pn (500 k,2400 b/s)	-168 dBW
--- Eb/No	+17 dB

Mode-L links (selected for RUDAK)

Downlink	P-Tx-S/C (5W)	7 dBW
	Gant-S/C	9 dBi
	Link-loss (apogee)	-177dB
	misc. losses in S/C and link	-3 dB
	margin	-7 dB
	Gant-ground	+10 dBi
	Power arriving at ground Rx	-161 dBW
	Pn (400 b/s,1000K)	-173 dBW
	--- Eb/No	+12 dB

Uplink	P-Tx ground (12 W)	+11 dBW
	Gant-ground	+15 dBi
	link-loss	-187 dB
	misc. losses	-3 dB
	margin	-7 dB
	Gant-S/C	+13 dBi
	Power at S/C Rx	-158 dBW
	Pn (2400 b/s, 500 k)	-168 dBW
	--- Eb/No	+10 dB (OK with PSK)

A MORE WATCHFUL WATCHDOG FOR MICROCOMPUTERS

Paul Newland, ad7i
Post Office Box 205
Holmdel, New Jersey 07733

Introduction

Many hardware/software watchdog timers consist of a software routine that repetitively triggers a hardware retriggerable monostable. If the monostable ever times out, the computer is reset. This technique, although useful, is not extremely reliable under most software/hardware insane conditions. This paper discusses an alternative approach that may prove to be more reliable under various fault conditions.

Discussion

The improved system offered here involves the use of a serial pseudo-random number (PN) generator, implemented in software, and a serial pseudo-random number detector, implemented in hardware. During the execution of the system program, a small portion of the CPU's time is allocated to generating another bit of the PN sequence and dumping it (and an associated clock signal) out a parallel port to the hardware PN detector.

When the clock lead from the parallel port goes high, the hardware checks to see if the data bit is a valid part of the PN sequence. If so, it does nothing. If it is incorrect, it resets the computer and doesn't allow the PN detector to issue another reset for a pre-determined period of time. This time delay is necessary to allow the computer to restart all processes and get the PN generator and PN detector back into synchronization.

While this PN checking is going on, additional hardware checks a tap from the hardware shift register to ensure that data is changing state occasionally. A failure of data change would indicate that either the computer is not clocking the external hardware or the computer is "fooling" the PN detector by giving it a degenerate PN sequence[1].

Implementation

Figure 1 shows a subroutine, written in Zilog mnemonics for the Z80(tm) processor, that implements a serial PN generator. The describing polynomial is:

$$Do = 1 \oplus Do\hat{}3 \oplus Do\hat{}7$$

where "Do" means the Data-output value for the current bit time and Do^n means the Data-output value "n" bit times ago. The "⊕" operator is the modulo-two sum (exclusive or) operator.

Every call of this routine generates a new bit output and an associated clock signal. Figure 2 shows hardware for the PN detector and the timing circuitry.

Conclusion

It is apparent from the discussion that any perturbation in the PN sequence (i.e., wrong bit at the wrong time, data stuck high or low, slow clock) will cause the reset signal to be issued. However, this "watchdog" can not protect against a computer malfunctioning in a way that causes the software-driven PN sequence to continue being generated. Programmers who might rely on a system such as this to protect against software "getting loose" should carefully consider where and when the PN generator routine is called.

Notes

[1] Typically, for PN detectors implemented using shift-registers and exclusive or gates, the detector can be fooled into thinking that the PN sequence received is correct if that sequence is either an all ones or all zeros pattern, depending on the construction of the detector. For this discussion, consider a degenerate PN sequence to be one of either all ones or all zeros.

```
PNBIT:  ; generate another bit of the PN sequence
        LD      HL,PNREG        ; pt to PN register
        LD      A,(HL)          ; get contents
        CP      0FFH            ; see if degenerate sequence
        JR      NZ,PNB10        ; jump if not
        XOR     A               ; zero contents if degenerate
PNB10:  AND     44H             ; check taps, clear CY
        JP      PO,PNB20        ; jump if next bit is to be zero
        CPF                     ; set CY to one
PNB20:  LD      A,(HL)          ; get old contents of register
        RLA                     ; shift in next bit
        LD      (HL),A          ; save new register contents
        AND     1               ; assume data is B0, clock is B1
        OUT     (PORT),A        ; set up data, clock is low
        SET     1,A             ; get ready to clock it
        OUT     (PORT),A        ; same data, clock now high
        RET                     ; that's all folks!
```

FIGURE 1: SUBROUTINE FOR GENERATING A PN SEQUENCE

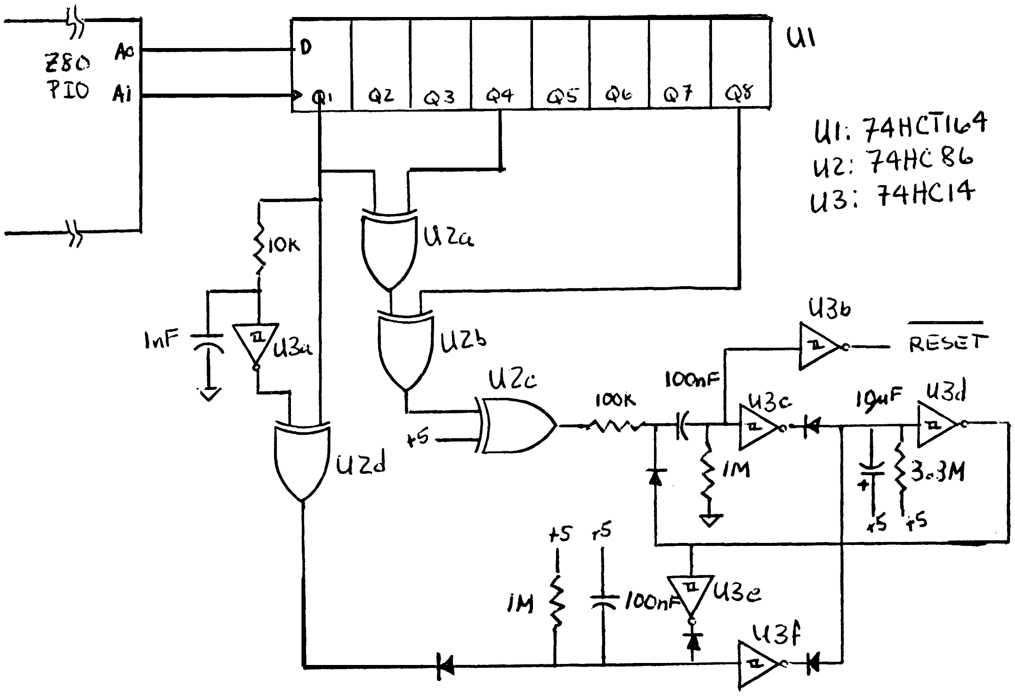


FIGURE 2: HARDWARE IMPLEMENTATION OF PN DETECTOR AND TIMERS

A FEW THOUGHTS ON USER VERIFICATION WITHIN A PARTY-LINE NETWORK

Paul Newland, ad7i
Post Office Box 205
Holmdel, New Jersey 07733

Abstract

This paper presents an idea for verifying that a user within a party-line network is who he or she claims to be. The idea assumes that the channel is a party-line and that potential intruders will monitor authorized communications and may attempt to masquerade as authorized users. No attempt is made to encrypt the authorized user's data for transmission over the party-line.

Introduction

Verification of a users identity to a host, historically, has been a major problem. At the lowest level of protection, many hosts require the user to "login" before he or she can make use of any of the system's protected resources. Login procedures, making use of passwords, are useful on most systems because it is assumed that a potential intruder will not be able to monitor easily the login process and determine someone else's password.

However, in an open system interconnected by unencrypted data communications radios (i.e., a party-line), the login process using passwords is ineffective because any casual observer can, by monitoring the radio channel, determine any other user's password. Also, once the user is logged in, there is nothing to prevent someone else from masquerading as the authorized user.

Offered below is an alternative approach to passwords for open, unencrypted, communications with a host. The method assumes that the user wants only to give commands to the host, not to do file transfers. However, with a few embellishments, the system could be expanded to include file transfers. The system would be as follows.

Proposal

When the host gives a prompt to the user, telling the user that she or he may send it the next command, the prompt will include a random number (RN) as part of the prompt. Each time a prompt is transmitted, a <u>new</u> random number is included. The host then encrypts the last RN transmitted using the key-for-the-day (KEY) previously agreed to by both the authorized user and the host; I will call this "Encrypted RN" the "ERN". The host would then place the ERN in storage for safe keeping and quick access. The user, on receipt of the new RN, would also encrypt it using the KEY (obtaining the same ERN if the KEYs are the same) and would save it, too, in a handy place. When the user wanted to issue a command to the host, it would give the command, as the host requires, but it would include the latest calculated ERN. When the host receives the command line, it compares the ERN received from the user with the ERN it calculated from the last RN transmitted as part of the prompt. If they match, the host assumes that the user knows the correct key. Thus, by returning the expected ERN, the user has proved to the host that he or she is authorized to access the system.

Vulnerabilities

This scheme is not without its problems. Because the key is changed only once a day, the range of numbers used for RNs must be large to ensure that RNs are not repeated during the 24-hour period. Also, the range of RNs must be large enough to make the probability of either determining the KEY or "guessing" the ERN sufficiently small. A RN of 64 bits should be large enough to overcome this problem. Another important consideration is the quality of the encryption technique. For Amateur Radio applications the DEA (that's Data Encryption Algorithm -- the software approach to DES) should be suitably strong.

This system is vulnerable to the sophisticated intruder who monitors the channel to determine the user's ERN and, before the user's packet transmission to the host is complete, the intruder jams the channel to block the user's packet from reaching the host. Next, the intruder sends his or her own packet with an illicit command and the ERN received

from the user. This scenario can be avoided by having the user transmit the ERN as the last item contained in the data segment of the packet.

Implementation

How might a system such as this be implemented? One way would be to include the host's 64 bit RN in the command prompt using hex notation and prefacing the RN with a special character sequence, perhaps composed of an unlikely sequence of punctuation characters so that a computer could identify it. Here's an example where the normal host system prompt is the word "yes?" and is prefaced by the RN.

:///:0FEB23178ED83A9A yes?

The user% terminal emulation program (or, possibly, packet controller (TNC)) would assume that a new RN follows the escape sequence ://: and that it is in hex notation. It would, using the key-for-the-day, encrypt the RN to create an ERN to be saved for future use. When the user gives a command to the host that requires the transmission of the latest ERN, the operator would tell the terminal program (or, again, the TNC) to substitute the ERN in place of an escape sequence. When the terminal program sees the escape sequence (such as ##), it would substitute the ERN (with a different preface than the RN) before transmission. An example of a command line with the escape sequence might be:

COFFEEPOT ON ##

The terminal program would substitute the escape sequence (##) with the last calculated ERN. An example:

COFFEEPOT ON /::/07EAF832B1A9E9A2

A method such as this would add about 20 characters 'of overhead to each packet that contained a command for a "protected" host. However, if the protection is adequate for the environment, that appears to be a small price to pay.

Extensions

Again, although the idea presented here is simple, it should prove adequate protection against the more-than-average hostile user who might try to disrupt authorized communications. Other extensions to improve its protection might include calculation of a checksum or a more complex data reduction technique across all data within the frame and use that as another dimension for encryption to create the ERN[1]. Certainly what has been presented here is only a first step. I encourage others to pursue this and additional methods of providing authentication within an open, party-line, network.

Notes

[1] Private communication with Mr. Philip Karn, ka9q, Bell Communications Research.

ANOTHER APPLICATION NOTE DESCRIBING A LOW-POWER RS232-LIKE INTERFACE

Paul Newland, ad7i
Post Office Box 205
Holmdel, New Jersey 07733

Abstract

A new and improved low power RS232-like interface is discussed. It features half the power consumption of the interface presented in a previous paper and uses fewer parts.

Introduction

In a previous paper[1] I showed a method of providing a low power RS232-like interface. That interface met all the stated requirements. However, for those applications that do not need full control of the receiver slicing point and the hysteresis, the interface described here may provide an alternative solution with reduced parts count that is acceptable for most applications. As an additional bonus, the power supply current drain is even less than that of the original interface.

Implementation

The output buffers (drivers), shown in Figure 1, remain functionally the same as those of the original. The current-limiting resistors in series with the output are not absolutely necessary but they are shown here for completeness. The reference point for the drivers is no longer buffered by an op-amp; the reference is set by 2 resistors instead of the previous one resistor and two diodes. A suitable op-amp would be the LM358 (dual) or LM324 (quad) for speeds up to 1200 bps. At speeds above 1200 bps, the slew rate of either type of op-amp exceeds the RS232 recommended values. For better performance at higher speeds, use a LM349 which has a slew rate five times faster than either the LM358 or LM324.

The input buffers (receivers), as shown in Figure 2, are either 74C14s or 74HC14s with the choice left to the user. The 74C14 requires less supply current and provides a larger hysteresis loop while the 74HC14 consumes more supply current (but less than 100 uA) and has a smaller hysteresis loop.

Because both the 74C14 and 74HC14 include clamping diodes at the input of each inverter section, a 100K series current-limiting resistor is all that is needed to protect against over-voltage input conditions. Additionally, to provide a default state when no signal is present at the receiver's input, either a 10K pull up (to +5) or pull down (to ground) may also be included.

Useful Additions

For those who are using a system that does not have a negative supply, a voltage inverter capable of supplying the necessary current for the output buffers is shown in Figure 3. It is capable of supplying up to 50mA without significant ripple.

For those who also need a regulated negative 5 volt supply, a spare section of an op-amp and a transistor can be used to provide good regulation. The resistor from the non-inverting input of the op-amp to ground is included so that the closed loop gain is greater than 5 (the LM349 is potentially unstable with closed loop gains of less than 5).

Notes

[1] Paul Newland, "A Low-power RS232-like Interface", Proceedings of the Third ARRL Amateur Radio Computer Networking Conference, April 15, 1984, pp 83-84.

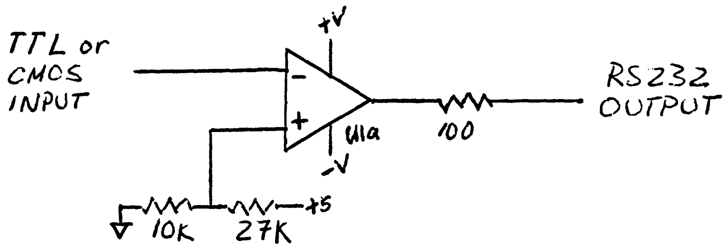


FIGURE 1 : OUTPUT BUFFERS

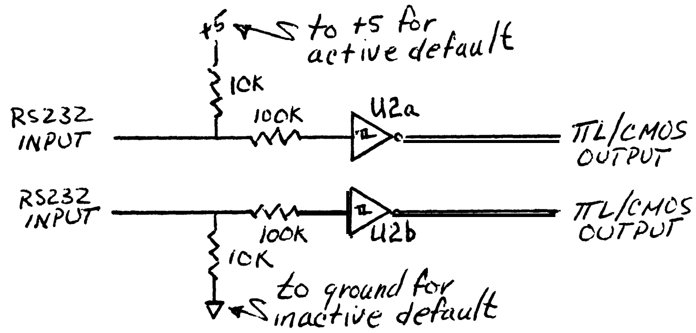


FIGURE 2: INPUT BUFFERS

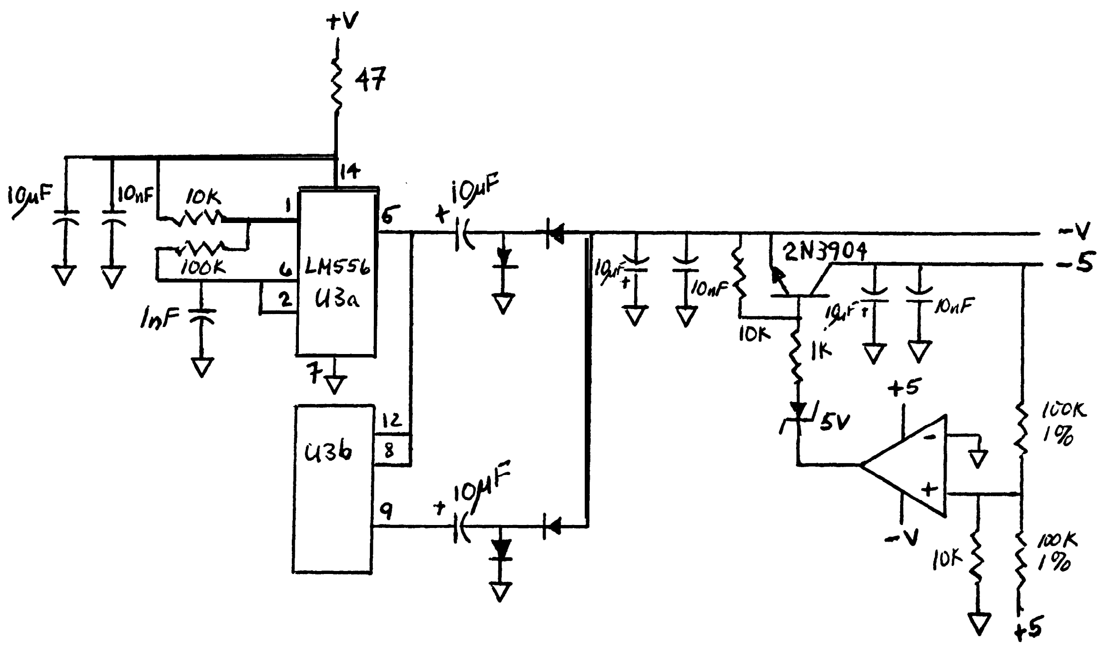


FIGURE 3: VOLTAGE INVERTER AND REGULATOR

The Realities of Packet Radio in the Amateur Radio Service, circa 1985
or
How to deal with a user base.

Harold E. Price, NK6K

Abstract.

The author postulates the existence of two major experimenter groups in amateur packet radio; those who experiment with data sent via packet radio and those who experiment with the way data is sent via packet radio. The problems of these groups in the face of 5000 or more packet users by the time of the 5th ARRL networking conference are discussed.

Let me preface this discussion by noting that some of the thoughts presented here were the result of group discussions during a meeting of the ARRL Ad Hoc Digital Communications Committee in 1984. At that meeting, the major discussion was layer three networking.

We have the privilege of witnessing the birth of a major force in amateur radio, one that may even have a lasting effect on its future. It is clear that the majority of technically minded junior and senior high school kids are now taking up computers as a hobby instead of amateur radio as in years past. If any of the plans being set in motion by the ARRL, magazines, and manufacturer's groups are successful in bringing new blood to amateur radio, the technology oriented newcomers will surely bring their computers with them. Packet radio will be a carrot to attract new blood.

In 1985, we will see a large increase in the amount of "press" the packet radio gets. 1984 was a good year, with major articles in the amateur press as well as such non-amateur publications as 'BYTE". There were also several peripheral mentions in PACSAT and UoSAT-11 articles in several areas; IEEE Institute, INFOWORLD, USA Today, Science, and others.

1985 will see several more articles is such places as IEEE Spectrum and a special issue of IEEE Communications. The biggest increase in packet's visibility will come from new manufacturers entering the market. The number of advertising pages containing packet equipment will double or triple in the next few months.

The bottom line is that packet will continue to grow at an increasing rate. It has grown from 240 to more than 2400 users in the last 14 months. It will at least double in the next year. As the growth of packet continues, so will the split between two groups, those who want to use the network, and those who want to build it.

There are those involved in packet radio who want to "play" with networks. Here the word "play" is not used as in Webster's definition 2(vi) 1c(2) ".... to behave frivolously", but rather as in 2(vi) 2b(2) "to move or operate in a lively, irregular, or intermittent manner". Those packeteers with the right stuff wish to push the edges of the envelope. They in particular, to judge from conversations that spring up at all gatherings where networking is discussed, wish to experiment with routing schemes. Zip codes, area codes, grid squares, zones, directions, random chance, casting of bones, any number of schemes are waiting to be tried.

Then there is the other group of packet users who wish to take the existence of a network for granted and get on with using it. Emergency nets, tornado spotting, traffic handling, newsletter distribution, public service events, earthquake detection (presumably by detecting a drop in traffic from California), and other data utilization topics are discussed in user's forums. The old term "appliance user" doesn't apply to these folks, anymore than it did to an oldtime op who didn't draw his own wire from an ingot to make a cat whisker for his crystal set. As we move into the future, the size and inner complexity of the basic building blocks changes. A good example of this is the WORLI store and forward message system. No knowledge of the inner workings of the AX.25 protocol was required to use a TNC for a building block to create something new, a way to get messages passed automatically between local area nets, and over HF, VHF, and Oscar 10.

Both groups need each other. A network must be designed and built to provide the services required by user community. And on the other hand, a network is no fun if

it has no users: how can you get enjoyment out of providing an elegant bottleneck avoidance algorithm if no one creates bottlenecks in the first place?

As the number of AX.25 TNCs grows, it becomes more difficult to make radical changes to them. The "TNC" will become a basic building block, it will have a set of assumed functions and a set place in the scheme of things, at least for a few years. As an example, when someone says "Grab your two meter HT and come help out in the marathon", there is almost no question that it will be compatible with all other two meter HTs. It will put out l-2 watts, be somewhere around 5 KHz deviation, and can be moved to any .005 MHz channel in the 144 to 147.995 range. There remain a few rockbound amateurs, just as there are still some TAPR 1.0 roms and Vl VADCG boards around, but you get the idea.

As it turns out, the requirement to build a network while not making any changes to the basic user TNC is a feature, not a bug. It forces a network design that isolates the inner workings of the long haul routing network from the general user. The result is a much smaller number of network routing devices. The smaller the number of devices and people involved, the more often changes can be made.

Figure 1 shows the architecture of a network that meets the goals stated above, requires no changes to the basic TNC, and reduces the number of devices with direct networking capability. In this diagram, users, denoted by boxes containing a 2 to show the highest protocol layer in use, connect to a network access node. Several users can connect at a time, and more than one frequency can be used. They establish a standard AX.25 connection with the device, and enter into a conversation with it to begin the connection process to some other network user. This is analogous to picking up a telephone handset. When you do so, you are "connected" to the telephone network. You tell the network who you want to talk to by entering an identification code. It is not necessary to know how the connection is made, only how to access the network (pick up the handset) and make a connection with another user (dial the number). The only other knowledge required is recognition of various error messages; busy, fast busy, a number of "We're sorry" messages, and a timeout on no action.

These same error messages will be present on a packet radio network access node. Since a network implementation will most likely be staged, the initial messages will be quite simplistic, perhaps even the familiar CONNECTED and RETRY COUNT EXCEEDED.

In Figure 1, the ---- connection lines are the standard AX.25 protocol. Lines marked as ====== can be any other protocol, although most planners have agreed to use AX.25 as the layer two protocol with various higher layers added on top. The important point is that the exact details of the connections between boxes marked III need not be known by the majority of packet users. As long as the interface between the user and the network access node (the boxes labeled III) stays the same, the network gurus can change the network at will so long as connectivity and throughput are maintained.

A final interesting point in figure 1 is the bottom left hand user. Since AX.25 is used for access to the network, simple digipeating can still be used by those on the fringes of local area nets. The added expense of a network access node is not required for users in very low activity areas.

Here is an example of the type of exchange that would take place between a user and the network access node. The actual data sent and received is in upper case, comments are in lower case and delimited by {}.

CONNECT NLA
*** CONNECTED TO NLA
NORTH LA NETWORK ACCESS NODE HERE.

{ A connection is established to a network access node. Node names need not be callsigns, the node could identify every 10 minutes with a UI frame.1

DI STATUS
{The user asks for status. Almost anything could be displayed here }

THERE ARE 5 OTHER USERS CONNECTED.
SB LINK IS UP
EASTLA LINK IS DOWN
SD LINK IS UP

{The list of connected network nodes is displayed. In the first networks, this will tell a user who he can expect to reach, based on his knowledge of the network. In the next 12 months, network nodes will be few in number and big in fanfare, so each local users will know the topology. A help file could be provided on the node for those who didn't}

CONNECT K6XXX @SFO
***CONNECTION ESTABLISHED.

{A connection via some number of III boxes is initiated and established.1

Once the connection is made, a transparent path is established through the network node, and data is passed directly to K6XXX, who is reached through the SFO

network node. An escape sequence similar to the transparent mode escape on the TAPR TNC or standard "smart modem" devices can be used to get back to the network node command level.

This method of network access allows for a staged implementation, something that is extremely likely to occur in the real world. When the network is simple, the network access program can be complex, allowing paths to be specified explicitly. As the network becomes smarter in routing, the connect command becomes simpler, until it is finally CONNECT **W3IWI.**

The intent here has been to quickly describe a way to implement a more complex network than is currently available while at the same time minimizing the impact of network construction on the majority of packet users. Many schemes are underfoot to provide network access devices, and the protocols to connect them. TAPR has agreed to work on the network node access protocol (the language used to **"talk"** to the node and get connected to someone at another node). Several people have suggested the use of the **A.3/X.28/A.29** protocols for TNC and network access node control. It is beyond the scope of this paper to go into depth on the exact access protocol or the network protocol itself, but it is hopefully not beyond the efforts of the amateur packet radio community. **Let's** get connected for Christmas.

```
+-------+                                                    +-------+
|       |                                                    |       |
|   2   |\                                                  /|   2   |
|       | \                                                / |       |
+-------+  \                                              /  +-------+
            \                                            /
+-------+    \ +-------+      +-------+      +-------+ /    +-------+
|       |     \|       |      |       |      |       |/     |       |
|   2   |------|  III  |======|  III  |======|  III  |------|   2   |
|       |     /|       |      |       |      |       |\     |       |
+-------+    / +-------+      +-------+      +-------+ \    +-------+
            /                                            \
+-------+  /                                              \  +-------+
|       | /                                                \ |       |
|   2   |/                                                  \|   2   |
|       |                                                    |       |
+-------+                                                    +-------+
    |
    |
+-------+
|       |
|   2   |
|       |
+-------+
```

Figure 1.

The Implications of Traditional Operating Practices for Amateur Packet Network Design

Gwyn Reedy, W1BEL

The Florida Amateur Digital Communications Association

Overview

Amateur packet network design relies heavily upon equipment and procedures developed for commercial service. This paper urges network designers to examine the anticipated uses of the network by amateurs, and modify commercial practice to accommodate the existing amateur population.

High Frequencies

The HF bands are generally characterized by long distance propagation which varies greatly by time of day, season, and sunspot cycle. Atmospheric noise is often severe. Tradition and the density of the user population prevent official channelization. (The variable nature of propagation would cause poor utilization of any strictly channelized frequency.) Standard operating frequencies for RTTY, slow scan TV, and other specialized modes approach a channel concept, and net operation is a method of channel allocation on a temporal basis. Only goodwill and general knowledge of those frequencies restrict interference. The lack of channelization makes possible the excitement and spontaneity of random operation which characterizes the HF bands. The practices of calling CQ or tuning for the calls of others enhance the variety of contacts which are possible. Many time-honored amateur activities such as DX chasing, contests, certificate seeking, and just plain "listening to the band" have flourished in the HF environment. However, point-to-point communication beyond daytime groundwave propagation distances on a reliable basis requires periodic frequency changes to compensate for changing propagation conditions. Even stations with power levels well beyond amateur levels must vary frequency in that fashion. Adaptive HF equipment exists that will determine and use the optimum frequency, but in general considerable operator skill and attention are required to optimize circuit throughput.

Very High Frequencies and Above

Operation on the bands above HF was little different from HF operation prior to the introduction of commercial FM equipment and procedures (and remains so on non-FM portions of the bands). Effective communication beyond the local line of sight area requires knowledge of propagation modes and current conditions. A smaller population of users and reduced propagation distances, combined with generous frequency allocations, reduces the chances of random contacts. Much activity is prearranged, but activities such as DXing, contesting, etc. flourish. Reliable point-to-point contact is easier to establish than on HF, using propagation modes which are line of sight or otherwise not dependent on the state of current band conditions.

The extreme popularity of FM operation on VHF is due to a number of factors. Improved audio fidelity, ease of tuning and extension of coverage by repeaters are some of the obvious ones. Amateur operating procedures have been modified to suit the mode, and are considerably different from those described previously. Calling CQ is discouraged (but has been replaced by the abhorrent practice of calling "QRZ" the repeater). Much less random communication takes place and while DXing repeaters is fun, it is considered poor form in populous areas. Many amateurs have adopted a 'home repeater' which they monitor regularly. This provides a high probability of making contact with a particular station if his 'home repeater' is accessed. I believe this ability to easily contact a desired station is one of the major factors in the popularity of FM operation, especially for base station operations.

Message System and Telephone Bulletin Board Operation

Many persons, including amateurs, that have personal computers are active in computer communication activities. The popular mode of contact for these hobbyists for computer to computer communication is the landline. (The voice call is certainly the most common means of telecommunications, but hams and computerists enjoy using their 'special equipment'), The natural two party limitation of conventional telephone connections and the single user capability of most personal computers have made the computer bulletin board system (BBS) a very popular method for non-real-time communication among multiple participants. It allows a person to leave messages or

computer files for another without regard to the availability of the receiver at the time of transmission. This same type Of capability has been implemented on the amateur bands by RTTY, ASCII, and AMTOR mailboxes or message system operations (MSO). The capabilities of many of these RF accessed systems are limited by the transmission speed of the mode used, interference on the designated frequencies, and the restrictions of the character set used.

Commercial Networks

The hardwired networks of terminals that communicate with host computers in commercial installations have provided amateurs with the hardware and link level protocols needed for packet networking. The environment on a hardwired terminal network is fairly constant with a known quantity and location of terminals. Communication connectivity is assured within the limits of the cable medium. All devices on the cable can hear all other devices, given that some time lags will require compensation. Thus each terminal may be assigned a fixed unique identifier which will be known by the host. All terminals are assumed to be available whenever the system is operational.

Packet Radio Networks

Several factors in amateur packet networks are significantly different from commercial networks: The unpredictable propagation of the RF environment, the sporadic nature of amateur operation, and the uses amateurs will make of the network.

In an RF environment all transmitters able to participate in a local network may not be able to hear each other. In addition amateur stations participate on a voluntary basis and cannot be assumed to be part of the network at any given time. As a result, if network connectivity is to be held at a high level, adaptive routing must be employed. This consideration is being satisfied in the level 3 and higher protocols now being developed by amateur network designers.

The factors I want to emphasize are those based on the possible uses of the packet network by amateurs. Commercial network users send messages or electronic mail to other network users. The addressee's identification is normally known by the sender, or his identity code can be found in a directory of some sort. There may be a general 'broadcast' facility which allows addressing all users or some subset of all users. Those modes of operation will be very useful to amateur network users. There will be great utility in sending out a message addressed to another amateur by callsign (and perhaps some other routing informabion) and having the message delivered automatically and rapidly to the recipient, or at least held in his local area until he activates his equipment. But over and above the tremendous value of that capability, there are many enjoyable, traditional things that amateurs want to do that have no commercial equivalent. Chasing DX, participating in contests, earning certificates, and listening to activity in other parts of the network are some examples.

Chasing DX on a packet network makes no sense, you say? DXing Digipeaters is good sport in Florida already, and just look at the way the Doctor DX game is gaining popularity. Surely if people can enjoy working DX with a computer program they will enjoy digital contacts with distant stations on a packet network. Contests? Last Spring the first contest was held using OSCAR 10, a shared, limited resource system. By all reports it was a success and did not adversely affect 'normal" operation.

Specific Activities That Should Be Accommodated On a Packet Network

I believe the following activities should be among those supported (in approximately this order of importance):

1. Network control and monitoring functions.
2. Remote computer access/operation.
3. Message/Mail delivery (short length).
4. File transfers (long length messages).
5. Remote access of network user and function directories.
6. Directional beaconing (broadcasting to a specific location).
7. Monitoring network traffic at a remote location.

The last three activities allow users to find out who is on the network and perhaps make a connection. These activities allow the network to support traditional amateur operating practice while giving priority to the dedicated information transfer functions that only the network can provide.

How might these capabilities be used by amateurs? Here are a few examples. A vistor in Florida for the winter wants to contact anyone in his hometown in the north. He may place a directed beacon (CQ) on the network that will be seen only by members of his home LAN. Similarly, he may enable a network process that will forward to his station any traffic addressed to his station on the hometown LAN. He may choose to monitor some portion of the activity in that LAN or any other LAN of his choosing. Limitations will have to be applied in this instance to reduce the load on network resources that would occur if all activity on a LAN were monitored. A desirable capability that substitutes processor loading for network loading would be

monitoring distant locations based on a keyword list. Only traffic bearing certain keywords (including callsigns) and related traffic would be forwarded to the monitoring station. Working network DX or earning the "Worked All LANS" award are procedural activities which become possible through directed beaconing and directed monitoring.

Summary

Packet radio is an exciting new mode of communication that is gaining immense popularity. It builds on the popularity of FM operation by providing a high (and growing) probability of automatic message delivery. It combines the pleasures and advantages of both computer and radio communication hobbies. Its appeal and usefulness can be enhanced for many amateurs if network design provides the capacity and facilities for random operation in somewhat traditional amateur fashion.

Ax.25 NET OPERATION IN THE CONNECTED MODE USING THE SOFTWARE APPROACH

Robert M. Richardson, W4UCH
22 North Lake Drive
Chautauqua Lake, N.Y. 14722

ABSTRACT:

This brief paper presents the means whereby an amateur radio net may be conducted using the AX.25 packet protocol with all stations in the connected mode. Use of a net control station connected simultaneously to all members of the net is described as well as window overlays on the video displays of all members of the net to display other net members packet information fields.

INTRODUCTION:

The title of this paper is an apparent contradiction of terms; i.e., how might one run a multi-station local net in the connected mode when a given station may be connected to only one station at a time? And, how a given station on the net connected only to net control, displays information fields from other stations on the net?

Is this your paradox or conundrum of the year for these proceedings, coach?

Not really, Gridley. There is no logical or rational reason why a given AX.25 protocol packet station cannot be connected to more than one station at a time. Since most stations use the hardware approach that only allows one station to be connected to another single station at a time, most packeteers presume this to be the case. Using the software approach to AX. 25 packet one may be connected to as many stations simultaneously as desired. Net control can use software to match the stations' on the net call letters/SSID and use separate number received (N/R) and number sent (N/S) counters appropriately. A window over the main menu allows net control to select to which net member his packets are addressed.

Ok coach, so net control can be connected to all the stations on the net simulataneuosly. Pray tell how all the stations on the net connected to net control can read each other?

Good question, Gridley. The answer is quite simple and an obvious one when you think about it. The stations on the net use a WINDOW overlay on their receive mode video display to display the TO - FROM - VIA call letters and info field of each info packet from net control and other stations on the net NOT directed to them. This is not a "fool-proof" system since it assumes that all stations on the local net receive signal levels at least as good as net control. It is not meant to be used by stations in "fringe" areas where umpteen retries are necessary to get a single packet through as this would disrupt net operation with unacceptable time consuming delays. All stations on the net would have the capability of filtering out interfering packets on the net frequency.

WINDOW SOFTWARE SUBROUTINES:

Windows were introduced by Xerox Parc many years ago and later implemented in many popular micros such as the Tandy 2000/1200/1000, Apple Lisa/MacIntosh, and IBM PC, and are truly fun and games to use when applied to AX.25 packet. For purposes of illustration, we will use the Radio Shack TRS-80 Models 1, 3, or 4 micros which all utilize the ubiquitous Z-80 microprocessor. Even with clock rates as low as the 2 MHz ballpark, (1.77 MHz for the Model 1), it is relatively easy to simulate concurrent processing. Not only is the hand quicker than the eye, a 2 MHz microprocessor is many thousands of times quicker than the eye.

Figure 1 is the main menu and figure 2 is the shift menu of the AX.25 dash 2 protocol software approach program. Many of the functions are automatically toggled ON or OFF, or NOW or NOT. Most input functions display a window over the menu and ask for the appropriate input in single or multiple window overlays.

Figure 3 illustrates a typical video display on the TRS-80 in the receive mode using the author's software approach program for the AX.25 dash 2 protocol. The top line shows that the proqram:

1. Is now in AX.25 protocol receive mode rather than Vancouver protocol.

2. Has been set up for 1200 baud. 300 or 600 baud for the HF bands may be selected from the shift menu if desired.

3. The NOW FORMAT option has been toggled on from the menu which recognizes and displays all carriage returns and line feeds on video.

4. The repeater function has been toggled ON from the main menu using WA1HDQ.

5. The program is in the NOW connected mode that was selected from the main menu.

Pressing shift 'N' displays the window shown in figure 4, This allows the user to set the window shape and size to most any configuration that suits the user's fancy. Setting the window's shape and size also automatically sets the program constants for the window's video display and scrolling. The window video overlay is entirely independent of the primary video display. Figures 5 and 6 illustrate two of the many possible window configurations the user may set up using shift N.

When the NOW display other info fields in the now connected mode is toggled on, from the shift menu by pressing shift Y, the program displays the TO - FROM - VIA call letters of stations the user is not connected to in the window's top line, and the info field in the lines below, within the window.

The user now has two choices that must be selected earlier. The first is to remove the window and restore the receive mode video display to normal by pressing ENTER. The second is to hold down the BREAK key until he or she has read the info field in the window, and then restore normal receive mode video by releasing the BREAK key. If he or she decides to do nothing, then the window is replaced by the normal receive mode video after a 2 1/2 second time delay.

The shift menu in figure 2 offers another function associated with the window display; i.e., the window filter that may be toggled ON by pressing shift K. Again, the user has two choices that must be selected earlier. The first is to filter out all info fields from up to 8 stations whose calls are input from the keyboard. The second is to display the info fields in the window from stations whose calls are input from the keyboard. Up to 8 stations calls/SSIDs may be input.

You may have noticed in figure 2 one last option that concerns the window function, namely shift F for NOT or NOW display calls in the connected mode. When toggled on, this displays a mini-window only one line high with the TO - FROM - VIA calls as shown below. If no repeater is being used the VIA displays "DIRECT."

```
TO WA1ABC FM WA1XYZ VIA WA1HDQ
```

The feature above is useful for non-net operation since the filter function may also be invoked. Forinstance, if you are awaiting a call from another station while working a station in the connected mode, he or she can let you know they are available by sending any variety of info frame, numbered or unnumbered. This feature displays the window for 1 1/2 seconds and then restores normal video. If you toggled the shift K filter function ON and input WAIXYZ, the only time a window is displayed is when WAIXYZ transmits an info frame. If you wish to change the response to only connect requests, it may be done in a few seconds using the edit/modify mode. Only one byte need be modified.

MULTI-STATION SIMULTANEOUS CONNECTION FOR THE NET CONTROL STATION'S PROGRAM:

Writing the subroutines for the multi-station simultaneous connection capability is another fun and games endeavor using the software approach. After the received frame is CRC checked in a few microseconds, the program progresses to test forward, and thence on to compare call letters. Here the compare subroutine can match up to 20 calls/SSIDs that you may input from the keyboard or conversely, automatically loaded into memory during net check-in time when each connect request is received and acknowledged.

Each call/SSID entered into the memory call list queue in the program also automatically creates separate N/R and N/S counters for each call, We have arbitrarily allocated enough memory for the net's call letters list and N/R + N/S counters to handle up to 20 net participants which seems adequate for most packet nets. As such, whenever you as net control are ready to transmit a packet the window overlay shown in figure 7 appears superimposed over the main menu. Pressing any key from A through T directs that packet to the station selected in the window with the correct N/R and N/S count for that particular station.

What if QRM or whatever causes one of the net members to miss a packet that net control received ok? Has it gone to never never land where scrolled off video bytes go?

Of course not, Gridley. Net control has the option of going into memory, lighting the blinking cursor, placing the cursor over the beginning of that packet, and then re-transmitting it if he or she chooses to do so.

CONCLUSION:

A considerable portion of the foregoing is mostly a "thinking out loud" aberation of the author. Most of the features EXCEPT for the multi-station connection subroutine have already been written and incorporated into our AX.25 dash 2 program that is available on disk for the Model 1, 3, or 4 (Mod 3 mode) from Richcraft as listed under REFERENCE (1) at the end of this paper.

The important point we are trying to make is that there are many routes to achieving

net operation, even within the meets and bounds of the excellent AX.25 dash 2 protocol. Need one change or modify the AX.25 dash 2 protocol for net operation? No, I personally think it is not necessary to do so. Surely, some brilliant packeteer with much more creative ability than the author will come up with a better and much more logical solution that does NOT require modifying the protocol. We have considered using the "star topology" and having each net member ACK each re-transmitted info packet, but the time involved with large number of net members precludes this rather pedantic approach.

The competition for amateur radio packeteers amongst the two major suppliers of packet radio systems using the hardware approach, primarily TAPR/AEA and GLB Electronics is hot and heavy. We hope that our old friend W2EUP of GLB Electronics fame will forgive us for placing his software approach in hardware EPROM, in the hardware category,

Our book, (1) "Packet Radio Using The Software Approach - AX.25 Protocol" is now in its 5th printing. No Gridley, we do not print 10,000 copies at each time, but the number of packeteers out there in the vast wasteland of computer land that are successfully using the software approach is growing by leaps and bounds.

We have modified the AX.25 dash 2 program to allow the use of a number of I/O ports and/or memory mapped addresses available to the Model 1, 3, or 4 TRS-80 user. Using the supplied CALTRS/CMD program on disk, the user may customize his or her program with NO knowledge of assembly language programming. CALTRS/CMD automatically loads the program with the users call letters in the appropriate places and then asks, "INPUT I/O DESIRED." I/O may be specified for the following:

1. A port encoder and decoder for ports zero through 127.

2. The line printer I/O memory mapped address for the Model 1 or the line printer port for the Models 3 and 4.

3. The ports used by the RS-232 interface (not the RS-232 function).

The latter two I/O selections eliminate the necessity for a separate I/O port encoder and decoder. Only a low cost home-brew EXAR 2206/2211 modulator/demodulator is required that costs approximately $11 to $15 total.

After the program is customized it is DUMPed to disk one time only, using the DOS dump command. Specific dump instructions for TRSDOS 1.3, 2.3, NEWDOS+, & NEWDOS 1.0 are included. The program automatically recognizes which Model it is in and loads the appropriate timing constants into the transmit and receive subroutines.

```
                 ENTER OPTION DESIRED ? _

CHANGE ADDRESSEE CALL LTRS = A   WA1XYZ CONNECT REQUEST CQ  = B
NOW CONNECTED TOGGLE       = C   WA1XYZ DISCONNECT REQUEST  = D
SEND PACKETS FROM LO-MEM   = E   WA1XYZ CONNECT ACKNOWLEDGE = F
INPUT FRAMES/PACKET LO-MEM = G   THIS IS AX.25 PROTOCOL     = H
BACKOFF DELAY TOGGLE OFF   = I   AUTO CONNECT TOGGLE OFF    = J
NOW IN LOWER CASE MODIFY   = K   AUTOMODE BEACON TOGGLE OFF = L
DISPLAY/EDIT MEMORY PAGE   = M   W2EUP - GIL BOELKE MESSAGE = N
NOW FORMAT VIDEO TOGGLE    = O   SEND ALPHABET TEST MESSAGE = P
VIA WA1HDQ/P REPEATER ON   = Q   SET OPENING FLAG LENGTH    = R
CHANGE REPEATER CALL LTRS  = S   INPUT/XMIT NORMAL INFO = V & T
CLEAR NON-PGM MEM 16K-62K  = U   INPUT/XMIT UNNUMB INFO = V & W
ABORT LOW-MEM PAK SEQUENCE = X   NOT CONEK TO OWN STATION   = Y
SHIFT MENU                 = 1   SET RE-TRY IN CONNECT MODE = 2
SEND WAIT REQUEST (RNR)    = 3   SEND CLEAR WAIT (RR)       = 4
```

Figure 1

```
                     SHIFT MENU ? _

SEND DM DISCONNECTED MODE  = A   BOOT DOS READY              = B
SEND FRMR FRAME REJECT     = C   MOVE HI-MEM TO LOW-MEMORY   = D
EDIT/MODIFY INSTRUCTIONS   = E   NOT DISPLAY CALLS IN CONEK  = F
DO NOT TEST RCV KYBD INPUT = G   ENABLE TEST RCV KYBD INPUT  = H
SEND MORSE I.D.            = I   SEND BEACON MESSAGE MANUAL  = J
NOT FILTER WINDOW OVERLAY  = K   DISPLAY LOW MEMORY @ 16384  = L
DISPLAY RECV PACKS @ 53248 = M   MOVE WINDOW CONNECT MODE    = N
OSCAR 10 CALL/HANDLE LIST  = O   MOVE PROGRAM TO LOW MEMORY  = P
SAVE HI-MEM ON DISK        = Q   LOAD DISK FILE TO HI-MEM    = R
TRANSMIT BAUD RATE SELECT  = S   TEST OTHER STATION STATUS   = T
CLEAR HI-MEMORY 53248 +    = U   SEND MORSE FROM KEYBOARD    = V
RECEIVE AX.25 PROTOCOL     = W   RECV VANCOUVER NOT CONNECT  = X
J/O GOES HERE                    NOT DIZ OTHER INFO CONNECT  = Y
       NOTE: SPACE BAR IN RECEIVE MODE = RESEND LAST PACK
```

Figure 2

```
RECEIVE AX25 = 1200 BAUD ----> NOW FORMAT  RPTK ON   NOW  CONNECT
The title of this paper is an apparent contradiction of terms;
i.e., how might one run a multi-station local net in the con-
nected mode when a given station may be connected to only one
station at a time? And, how a given station on the net connect-
ed only to net control., displays information fields from other
stations on the net?

Is this your paradox or conundrum of the year for these proceed-
ings coach?

Not really, Gridley. There is no logical or rational reason why
a given AY.25 protocol packet station cannot be connected to
more than one station at a time. Since most stations use the
hardware approach that only allows one station to be connected
to another single station at a time, most packeteers presume
```

Figure 3

```
RECEIVE AX25 = 1200 BAUD ----> NOW FORMAT  RPTR ON   NOW  CONNECT
The title of this paper is an apparent contradiction of terms;
i.e., now might one run a multi-station local net in the con-
ncctcd node when a given station may be connected to only one
station at a -------------------------------------- net connect-
ed only to n [TO WA1ABC FM WA1XYZ VIA WA1HDQ     ] from other
stations on  [ ARROW TO MOVE WINDOW              I
             [SHIFT RIGHT/DOWN ARROW EXPAND      I
Is this your [SHIFT LEFT/UP ARROW CONTRACT       ] hese proceed-
ings coach?  [BREAK RETURN TO MENU               I

Not really, Gridley. There is no logical or rational reason why
a given AX.25 protocol packet station cannot be connected to
more than one station at a time. Since most stations use the
hardware approach that only allows one station to be connected
to another single station at a time, most pncketeers presume
```

Figure 4

```
RECEIVE AX25 = 1200 BAUD ----> NOW FORMAT RPTR ON   NOW CONNECT
The title of this paper is an apparent contradiction of terms:
-----------------------------------------------------------------
[TO WA1ABC FM WA1XYZ VIA WA1HDQ                                 ]
[                                                               ]
[                                                               ]
[                                                               ]
-----------------------------------------------------------------
Is this your paradox or conundrum of the year for these proceed-
ings coach?

Not really, Gridlcy. There is no logical or rational reason why
a given AX.25 protocol packet station cannot be connected to
more than one station at a time. Since most stations use the
hardware approach that only allows one station to be connected
to another single station at a time, most packeteers presume
```

Figure 5

```
RECEIVE AX25 = 1200 BAUD ----> NOW FORMAT RPTR ON   NOW CONNECT
------------------------ is an apparent contradiction of terms;
[TO WA1ABC FM WA1XYZ   ] a multi-station local net in the con-
[                      ]  station may be connected to only one
[                      ] how a given station on the net connect-
[                      ] displays information fields from other
[                      ]
[                      ]
[                      ] conundrum of the year for these proceed-
[                      ]
[                      ]
[                      ] ere is no logical or rational reason why
[                      ] acket station cannot be connected to
[                      ]  a  time.  Since most stations use the
[                      ] nly allows one station to be connected
------------------------ n at a time, most packeteers presume
```

Figure 6

```
                   ENTER OPTION DESIRED ? _
                   -------------------------
CHANGE ADDRESSEE  [NET MEMBERS CHECKED IN] NNECT REQUEST CQ = B
NOW CONNECTED TOG [    APRIL 1, 1935     ] SCONNECT REQUEST = D
SEND PACKETS FROM [WA1ABC = A  WA1KLM = K] NNECT ACKNOWLEDGE = F
INPUT FRAMES/PACK [WA1BCD = B  WA1LMN = L] X. 25 PROTOCOL    = H
BACKOFF DELAY TOG [WA1CDE = C  WA1MNO = M] ECT TOGGLE OFF    = J
NOW TIJ UPPER CASE [WA1DEF = D  WA1NOP = N] BEACON TOGGLE OFF = L
DISPLAY/EDIT MEMO [WA1EFG = E  WA1OPQ = O] IL BOELKE MESSAGE = N
NOW FORMAT VIDEO  [WA1FGH = F WA1PQR = P] ABET TEST MESSAGE = P
VIA WA1HDQ/R REPE [WA1GHI = G  WA1QRS = Q] NG FLAG LENGTH    = R
CHANGE REPEATER C [WA1HIJ = H WA1RST  = R] T NORMAL INFO = V & T
CLEAR NON-PGM MEM [WA1IJK = I WA1 STU = S] T UNNUMB INFO = V & W
ABORT LOW-MEM FAK [WA1JKL = J  WA1XYZ = T]  TO OWN STATION  = Y
SHIFT MENU        ------------------------- Y IN CONNECT MODE = 2
SEND WAIT REQUEST (RNR)  =  3   SEND CLEAR WAIT (RR)        = 4
```

Figure 7

<u>REFERENCE (1):</u>

Packet Radio Using the Software Approach-AX.25 Protocol. $22 postpaid US & Canada

disk: specify Model I or Model III TRS-80
$29 postpaid US & Canada
(book above is required)

from: Richcraft Engineering Ltd.
#1 Wahmeda Industrial Park
Chautauqua, New York 14722
phone: (716)-753-2654

COMPUTER NETWORKING IN JAPAN 1985 - ONWARDS AND HOW BILL GATES' (MICROSOFT) CREATED THE FAR EAST & PACIFIC MSX STANDARD

Robert M. Richardson, W4UCH
22 North Lake Drive
Chautauqua Lake, N.Y. 14722

ABSTRACT:

The evolution, development, and implementation of the Microsoft MSX operating system in nearly 2 dozen models of microcomputers now being manufactured in the Far East and Pacific (FEP) is discussed along with their impact on packet radio on the amateur bands primarily in Japan, for the 1985 - onwards, time frame.

INTRODUCTION:

Gridley, here is your trivia question of the week. Why is Apple to microcomputer similar to Microsoft to software?

Gotcha coach. They are both the biggest in their respective fields.

Not really, Gridley. IBM's dividends exceed Apple's 2 billion dollar annual sales. Furthermore, there are a number of software houses that are as large or larger than Microsoft% $200 million annual sales. Try again, Gridley. Think back to the 1975 - 1976 era of yesteryear tall masked man.

Aha, you gave me a clue, coach. Both firms were started by rather young computer buffs working in garages, dormitory rooms, or the attic.

Right you are, Gridley. Steve Wozniak and Steve Jobs at Apple, and Bill Gates and Paul Allen at Microsoft were the leading perpetrators - founders, of these two outstanding firms.

So what does that have to do with the MSX missile program, coach?

Absolutely nothing, Gridley. The 'MS' stands for Microsoft and the 'X' stands for the Microsoft extended Basic interpreter. MSX is what this short paper is all about, plus its impact on amateur packet radio in the Far East and Pacific basin, especially in Japan.

BY 1979, Microsoft's BASIC interpreter, first written by Gates and Allen circa 1975/1976, had become the defacto world standard BASIC, and still is today, Microsoft's achievements are really too numerous to mention, but two items particularly standout. The first and most obvious is MS DOS, the fundamental PC DOS disk operating system for the IBM PC. The second and not quite so obvious achievement was the Microsoft "Softcard" that turned the Apple II' into a real honest to goodness modern day micro by substituting the ubiquitous Zilog Z-80 microprocessor for the cheapy 6502 microprocessor with its very limited instruction set and minimal registers.

In the early 1980's, Bill Gates' foresight & precognition again came to the fore and Microsoft, called ASCII Microsoft Ltd. in Japan, developed the MSX operating system for microcomputers that utilize the Zilog Z-80A microprocesor (or its licensed counterparts manufactured in Japan). The idea here was an extremely sound one; i.e., any microcomputer using the licensed MSX operating system would have complete software program compatibily with any other microcomputer using the licensed MSX operating system. Beyond software compatibility, a further goal was to have completely interchangeable peripherals.

In addition to the usual line printer, light pen, etc., peripherals, they also include stereo sound, video disk, and video cassette, plus optional growth capability via the MSX DOS (disk operating system) to use the new 360K byte (formatted) 3 1/2" floppy disks now being manufactured in Japan. Microsoft provides the MSX disk operating system on a chip which is built into the floppy disk drive adaptor.

FUNDAMENTAL MSX OPERATING SYSTEM SPECS:

Microprocessor	: Z-80A 4 MHz clock
MSX ROM	: 32K bytes standard
MSX RAM	: 32K average - up to 64K
Video control	: Texas Instr. TMS-9918A
Text display	: 32 characters X 24 lines
Graphic display	: 192 X 256 dots 16 colors
Sound generator	: General Instr. AY-3-8910
Audio range	: 8 octaves, 3 tone chord
Peripheral chip	: Intel i8255
Line printer	: Centronics parallel port
Cassette	: 1200 bps low 2400 bps hi

This sure looks like a fun and games specification to me!

Yes, and no, Gridley. The MSX spec is aimed at the universal "home computer" market. Notice we did not say "personal computer" market as there is a substantial difference between the two. It is obvious from the MSX text display specs that it is

designed to work with a standard home television set serving as the video display as 32 characters by 24 lines is the maximum that a standard home TV set's video bandwidth can handle. Video output is to the TV set's video input or via an optional TV channel RF modulator. Analog Red Green Blue (RGB) output is another option.

It is worth noting that the majority of MSX micros offer RAM expansion to 64K bytes, so undoubtedly virtually all of them utilize bank memory switching between ROM and RAM. The ROM may be plugged into the MSX cartridge slots, one or more, on all MSX micros. You should also note that virtually none of the MSX micros include or offer the RS-232 serial interface that the hardware variety of packet radio buff is so fond of using.

Another fascinating aspect of the MSX specification is what ASCII Microsoft Ltd.% Mr. Kazuhiko Nishi terms the "MSX Engine." This fabulous chip, soon to be manufactured by Toshiba combines the functions of the Z-80A microprocessor, the TMS 9918A video control, i8255 peripheral controller, and AY-3-8910 sound generator ALL on a single chip. The price tag is expected to be under $10 when mass procuction gets rolling.

MSX MICROCOMPUTER MANUFACTURERS:

Canon	(Japan)
General	(Japan)
Kyocera	(Japan)
Victor	(Japan)
Yamaha Nippon Gakki	(Japan)
Mitsubishi Electric	(Japan)
Sony	(Japan)
Pioneer Electronic	(Japan)
Matsushita Electric	(Japan)
Toshiba	(Japan)
Hitachi	(Japan)
Fujitsu	(Japan)
Sanyo Electric	(Japan)
Samsung Electronics	(S. Korea)
Gold Star	(S. Korea)
Daewoo Electronics	(S. Korea)
Limco Products	(Singapore)
Oric Electronic	(Singapore)
Command Module	(Hong Kong)
Radofin Electronics	(Hong Kong)

NOTE:
Another player in the MSX game:
Philips Eindhoven (Holland)
(UK, France, West Germany soon?)

Prices in US $ for MSX microcomputers vary from a low of about $150 up to a high of $380, depending upon options. The 360K byte 3 1/2" disk drives from Sony and Toshiba retail for an additional $330 which will surely decrease as volume goes up. Most of the above manufacturers, except Yamaha's Musical Instruments Division, do NOT plan to market their MSX micros in the U.S. Their major thrust will be in Europe and Japan, with a few targeting the Middle East with Arabic keyboards.

At the end of 1984 there were about 1/2 million MSX micros in use in Japan with the 1985 - 1986 forecast in the 1 to 2 million ballpark for Japan ALONE.

Ok coach, I am impressed. Now, let's get on with the packet story line which is what this discussion is supposed to be about.

Sorry Gridley, I got carried away by all these fascinating facts and figures.

JAPANESE RADIO AMATEURS 600,000 PLUS:

That's another impressive figure!

Yes indeed Gridley, especially when one considers that the first Japanese AMSAT bird with AX.25 packet capability is going to be launched in 1986, plus a number of forward thinking Japanese amateurs are already planning a number of AX.25 protocol packet repeaters for installation beginning the summer of 1985.

Hmmmmmmmm? Standard 32K plug-in-able ROM? No RS-232? Common standard keyboards? Common standard video displays? I have a great idea for the software approach to packet, coach!

Hold your horses, Gridley. We are working on the first dedicated packet ROM cartridge for an MSX micro for our Far East & Pacific distributor in Tokyo. Our 30 page instruction booklet will be translated into Japanese shortly.

S-I-L-E-N-C-E.

Sorry I stole your thunder, Gridley. Here are the rough details of this little joy and delight:

- End user price of the MSX packet cartridge will be the equivalent of $80 U.S.

- Requires 32K RAM which most all MSX micros have or exceed.

- The 1200 baud MSX cartridge has AFSK output to the user's transmitter via a cable and plug, a cable and plug for audio input from the user's receiver, and a cable and plug to the user's T/R relay.

- The 300 baud MSX cartridge is identical to the one above except the I/O is for 200 Hz shift on the HF bands.

- All plugs are configured for the Kenwood, ICOM, and Yaesu amateur transceivers.

- All features of the Richcraft dash 2 AX.25 protocol software approach are included. Instead of 2 menus, 4 menus are provided due to the 32 characters per line text display. Windows are provided over menus as well as on the

receive mode video display when toggled ON.

- Only half of the 32K available ROM is is used for the packet program and ancillary subroutines such as keyboard decoding, video display/control, etc. As such, there is plenty of room available for level 3/layer 3 implementation next year.

- The 32K RAM is partitioned into the following segments: 12K for processed received info frame storage, 12K for storing long files to be transmitted, 4K for packet assembly and disassembly, and 4K for variable storage. Those MSX micros with 64K RAM memory use the extra memory for expanded receive and transmit storage.

- Packet in living color is available if desired. Only the most frustrated art major would find the 16 colors inadequate. Zooming windows, color identification of TO - FROM - VIA and all sorts of Walt Disney animation is quite easy to accomplish.

When will we see the first MSX micros in the U.S.?

They are enroute if not already here Gridley, but do not look for them in your local computer wonderland store. The first MSX manufacturer that has stated publicly that they will give the U.S. market a try, is Yamaha who will distribute them through its Musical Instruments Division. Best visit your local music store if you want to see the Yamaha model YIS-503. Pioneer also plans to test the U.S. market later in 1985 with their model PX-7.

CONCLUSION:

AX.25 protocol packet activity is about to explode in Japan. The Microsoft MSX specifications are truly ideal for the Richcraft software approach to packet radio and allow operation at any baud rate from 300 on up through 2400 baud at extremely low cost. Just plug in an MSX packet cartridge and you are "on the air."

The author's non April 1, 1985 prediction:

"By 1987 there will be as many or more AX. 25 packet amateur radio stations in Japan as in the rest of the world combined."

Many U.S. amateur packet radio buffs will pooh pooh the MSX micro spec. Undoubtedly these are the same experts who pooh poohed Japanese designed and manufactured automobiles for the U.S. market a number of years ago,

ACKNOWLEDGEMENT:

Special thanks are due to Richcraft's astute Far East & Pacific distributor in Tokyo for providing the author with the reams of data that this brief article was based upon.

THE USUAL DISCLAIMER:

All errors and sins of omission are unintentional and strictly those of the author. All prognostications and predictions are strictly the author's best guesses as of December 1984.

Activity report of PARNET

Takemi Yamazaki, Masahiro Takeda, Motokazu Kashida,
Masa – aki Yonezawa, and Tadanori Harada

Abstract

We have been studying and working on digital communications in ham radio for the past year. The group activities are discribed in this paper.
In section one, there are member's profiles, group's aim/policy and results of our investigations. In section two, the hardware of the prototype is described. In section three, the software descriptions and in section four, prospect in the future is written.

Introduction

Members of PARNET (Packet Amateur Radio Network) are JI1BXM, JA3FGN, JA1MIR, JH1UWU and JE1WAZ. Members are either electronics engineers, or specialists of software and hardware.

Objectives

We gathered to achieve common goals of the project. Those are:

A)to study the method to connect our transceivers with our personal computers.

B) to establish a computer network through ham radio.

C) to construct a central station for the net – control which has many useful functions with large capacity of memories.

D) to connect our central station to other data networks, JAS- 1 Satellite repeater, AO – 10 and so on.

E) to provide the cental station as an open repeater for world-wide Ham stations.

F) to propose a new packet protocol which fits to the circumstances in JA, and to the mobile radio, if possible.

Current status

First, several modulation alternatives were investigated, and frequency spectrum was checked by using a computer. And we get the result that the FSK modulated by base- band is good for high baud rate. But it is not easy for a user to reconstruct or to buy a tansceiver for the special purpose. So we made a plan to use the AFSK modulation.

We consider that AX.25 is the standard to connect the worldwide network via a satellite repeater. So we select AX.25 as a protocol to hold a compatibility at the first step, We chose a TAPR-TNC as a compatibility checker and our software is under development through actual data exchange between ours and TAPR's.

On the other hand, we believe that 'easy to make' is important for a hardware. So our hardware is so different from TAPR's and has few tuning point.

We decided that our TNC supports the lower level protocols only. Higher level protocols (level 3 and above) should be handled by personal computers. And the software is designed to be able to change the structure to support yet another protocol which matches the conditions in JA and to the mobile radio in future.

Figure 1 – figure 4 show the result of the calculations of the power spectrum in 1200 baud AFSK modulation. 3 dashed lines in each figure indicates the bandwidth which covers 90%, 95% and 99% of power spectrum, respectively. FIG.3 and 4. show the power spectrum of Bell 202 and MSK respectively.

Hardware

Design goal of our prototype hardware was to simplify the circuit, to make it a small single board, and to reduce tuning points.

A block diagram is shown in Figure.5 where;

CPU: Intel Z80A	CLOCK 2.46 (4.915212) MHz.
RAM: 6116 X3	6 K Bytes (including battery backed up RAM 2 KB).
ROM: 2764 X2	16 K Bytes.
SERIAL PORT: 18251	For VDT or Computer.
HDLC CONTROLLER: I8273	Digital PLL circuit on chip.
MODEM: Am7910	Programable universal MODEM.
PTM: 18253	Genarate baudot rate clock.

The firmware on ROM controls these LSIs and electrical switches. We planned that the TNC system handles Physical layer and Data Link layer. The Network layer and higher layers are handled by a computer. (See

Figure 1 Power Spectrum

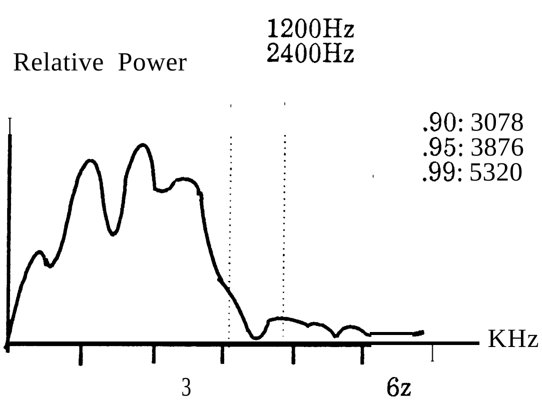


Figure 2 Power Spectrum

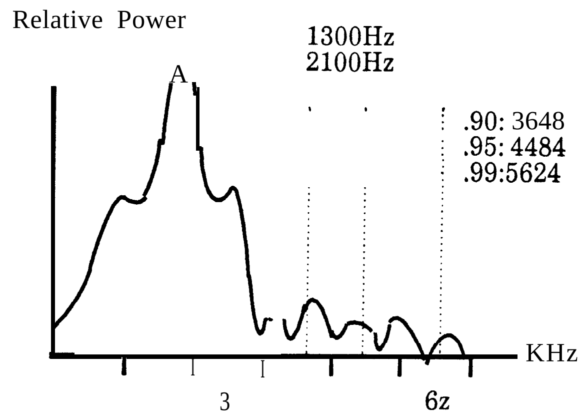


Figure 3 Power Spectrum

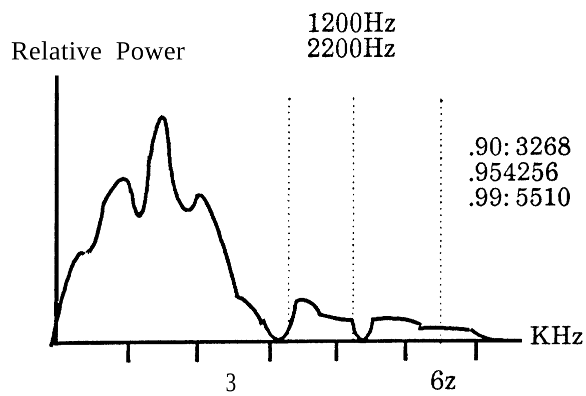


Figure 4 Power Spectrum

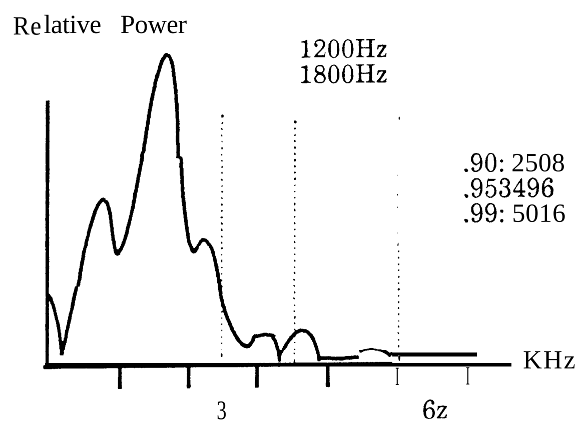


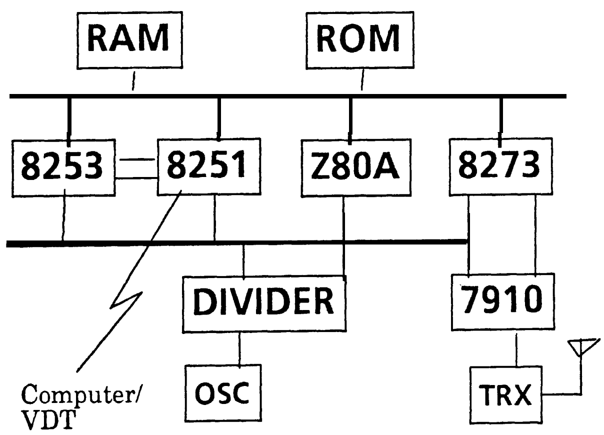
Figure.6) Z−80A processor has enough performance to deal with data and to control the circuit for handling Level 1 and Level 2. RAM area is 6 KB and it is enough to store several packets. Back -up circuit using Ni − cd battery is available to keep parameters set up by a user in static RAMs. SERIAL PORT uses RS-232C standard which is very popular so that the software for interface between the TNC and Computer is simplified. HDLC CONTROL Section is characterized by its Digital Phase Locked Loop (DPLL) circuit . There are hundreds of bits in a packet, so unsynchronized clock causes fatal receiving error. DPLL circuit regenerates clock from receiving data. So it is able to communicate on bit oriented data stream without a clock line. MODEM block is wonderful. There is no tuning point. It generates phase continuous AFSK audio signals, and meets to BELL 103/113/108, BELL 202, CCITT V.21 and CCITT V.23. To meet TAPR TNC, it is programmed to BELL 202.

Software in PARNET TNC

Software architecture

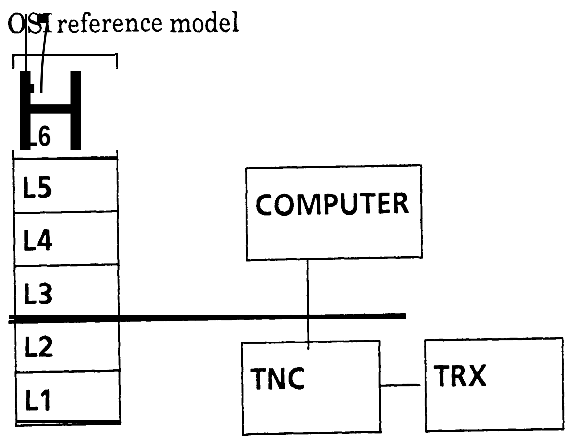
TNC has to execute the following jobs simultaneously.

Figure 5 Block diagram of TNC



A) Sequence control block: transfers strings (send − data), converses with the terminal, and controls the internal state.
B) Protocol control block: controls the packet header, manages the packet sequence number, and controls the transceiver PTT.

Figure 6 Layer handling



In case of the fact that a single CPU executes more than one task simultaneously, 'REAL TIME MONITOR' are often used as the operating system. But on the development of our TNC, we made the simple language for multi- task execution, not to get down the memory availability, and we use it in order to program the PROTOCOL CONTROL BLOCK.

This language manages the sequence number (label) like BASIC. The control of the routine is always interrupted at the beginning line of the loop, and the control transfer among concurrent procedures is occurred, to avoid possible dead − lock, which causes the violation of the correct sequence control.

The program is converted into language C using a preprocessor which is also written with language C. Since the SEQUENCE CONTROL BLOCK can be programmed just as the main routine, the efficient programming can be expected.

We use an NEC PC -8000 system and C compiler on the CP/M for the development of the software.

TNC Mode

PARNET TNC has two modes. One is the conversation mode on which we can converse with other pepole by using ASCII characters just like RTTY, and the TNC can be controlled by human readable command. The other is the binary mode where binary files can be

transferred. In the binary mode, the conversation and control are achieved by using ESCAPE SEQUENCES, and it is available for a computer to control the internal state sequences.

Layer assignment

The repeater addresses are set in each packet header and included in the layer−2 on AX.25 recommendation. But Japanese authority allows no digipeater at the present time. So the function of the digipeater address specification is not required in JA. Moreover, even if the digipeater was allowed, it should be required that the data flow path information must be allocated in level 3. It is the role of net work layer rather than link layer to control message forwarding. This decision gives us another advantage of realizing shorter packet during the test operation. The reason why we want to reduce the packet length is that the data communication is always (much frequently !!) disturbed by some stupid ham radio guys.

On this layer assignment, the data flow path can be also instructed by any user with a high level algorithm in the host computer.

Examination in the future

Our major activities, so far, were concentrated on realizing a TNC which is compatible with AX.25. We have completed a TNC which could be utilized to practical use. We are now planing to examine the various applications of the TNC to upper layers and also to look for a method of high−speed transmission, accompanied with good quality.

The items which we are looking at are as follows.

To desian and build a comouter for center station.

The idea of the computer is to have the main memory with the capacity of 1 to 2 M Bytes and a hard disc device with more than 20 M Bytes. We would like to use UNIX as OS and release this facility to common users.

To utilize error correction code

We would like to test the error rate in both with ECC and without ECC statistically, and if the difference is significant, examine the potential of realization of ECC with current TNC. We consider that ECC is necessary to relieve the interference in Japan (ie, jamming). We are looking at the realization by implementing micro−programming and if it is not feasible, we may have to introduce dual processor system.

To examine modulatina methods to enable hiqh - speed transmission

We want to realize the transmission at 12 Kbps to achieve high− efficiency and high− fidelity and implement a low pass filter with a duobinary square−root characteristics to reduce the out −of− band radiation, and the modulator has **4** transmission level. The demodulator will be a discriminator which is compatible with an current FM receiver.

AMATEUR PACKET RADIO IN IARU REGION 3*

Paul L. Rinaldo, W4RI
American Radio Relay League
Newington, CT USA 06111

INTRODUCTION

About 6000 radio amateurs throughout the world are now equipped for packet radio, and the number is more than doubling each year. Amateur packet radio was made possible by the introduction of the personal computer around 1975 and experimentation by radio amateurs since 1978. The common goal of these experimenters is to build a global network that will enable personal computers to exchange data rapidly and without errors. The network not only will handle conversational QSOs but will support many new services such as transfer of record messages, access to electronic data bases, transmission of videotex and facsimile images, and digitized speech. Packet radio is not simply a high-tech replacement for radioteletype; it is an automatic and reliable method of transmitting any digital information in short bursts.

While amateur packet radio had its start in the North America, it has now taken root in many countries throughout the world. In the Asia and Pacific areas, there is notable amateur packet-radio development in Australia, Japan and New Zealand. There are early signs that the Japanese Amateur Radio industry is looking very seriously at new products involving packet radio. Not only can they sell packet-radio products to amateurs worldwide, but there are many commercial and governmental applications for a full product line. In the United States, there already has been some spinoff of amateur packet radio into other radio services. Some current examples are: (a) U.S. Forestry Service use between personal computers in National Parks, (b) data communications with military severe-weather-surveillance aircraft, and (c) mobile communications with a large fleet of overnight courier vehicles.

Without a doubt, amateur packet radio is here and is here to stay. As in any new technology, it is not possible to predict its future twists and turns, or even its ultimate shape. Military or commercial projects can be organized in a top-down manner with clear objectives, financial control and technical direction. We amateurs do not have the benefit of a well-funded central authority to develop a network. Instead, we have individuals and groups with a great freedom of action and diversity operating with funds of piggy-bank proportions. The Amateur Satellite program was perhaps the first example of how amateurs learned cooperation and division of labor in a complex technical program that would serve all radio amateurs. Building a global packet-radio network will require the same synergism -- and more. We need the specialists to devise new protocols, software and hardware. We also need the organizers, facilitators and entrepreneurs to apply their efforts to the main stream without stifling innovation.

National Societies have a large stake in the outcome of packet-radio development in Region 3. Societies can help create an environment in which creative amateurs can flourish. Finding precious human and material resources is another role. Clearing away obstacles to packet radio in radio regulations is an important goal for National Societies and the IARU. Cooperation in international amateur packet-radio standards is another essential task for National Societies to ensure that the network is interoperable.

CREATING AN ENVIRONMENT FOR PACKET RADIO EXPERIMENTATION

Many countries in Region 3 already have centers of excellence -- areas with Amateur Radio clubs with technical interests, radio amateurs in the science and engineering professions, computers, laboratories, and universities. Chances are that amateur packet radio will sprout on its own in these areas. But for that to happen, there has to be some knowledge that packet radio exists and an individual who will be a spark plug or initiator. National Society membership journals, commercial magazines and newsletters can spread the word of packet radio's existence. In our experience, however, it takes a personal presentation by a packet-radio enthusiast to plant the seed. A demonstration of a working packet-radio system is needed by most amateurs for them to comprehend how it works and what it can do. A knowledgeable person can answer questions and clear up any misconceptions. Our experience is that this is sufficient to generate interest in these centers of excellence. Actually making packet radio happen in such areas usually depends on putting a packet-radio repeater (digipea-

*This paper was prepared for presentation at the International Amateur Radio Union Region 3 Conference in Auckland, New Zealand, November 13-17, 1985. As it contains information of interest to other areas, it is printed herein.

ter) on the air and getting at least two amateurs to start using it. From there, packet radio seems to be established and capable of further growth.

For packet radio to get started in high-tech areas, there must be continual access to equipment, software, and other system support. These things are readily available in the United States and are becomming available in other countries. In some countries, importing is a problem, and developing domestic sources may be possible. For example, you may wish to investigate local production of packet-radio equipment printed-circuit boards under license of a U.S. manufacturer. Some countries have the resources to developtheir own equipment; however, this is not a trivial task.

There will also be a continual need for the latest information about packet radio because it is rapidly developing. ARRL Headquarters keeps fully informed about packet-radio activities in North America and elsewhere, and is willing to share this information on a timely basis with other National Societies. ARRL publications (Gateway, QST, QEX and the Handbook) are the vehicles we regularly use to disseminate information about packet radio. In addition, we offer the proceedings of past ARRL Amateur Radio Computer Networking Conferences -- some 76 technical papers written by amateurs.

So far, I have covered only the simpler problem of how to get packet radio started in an area with essentially the right conditions. What can be done where the environment is not particularly conducive? It may be a long process and might have to waituntilthe day of mass availability of packet-radio systems. Perhaps the major manufacturers will eventually reduce the packet-radio controller to a single integrated circuit and make it an organic part of all-mode transceivers. Butwhatcan we do in the short term? The answer could lie in the centers-of-excellence concepts outlined above and encouraging such centers to set up networks between them, both nationally and internationally. Maybe that will attract others. "Packeteers" in the United States wishing to link from one city to another have been able to recruit another breed of ham -- the fellow who sees his role in life as putting VHF repeaters on the air. Newsletter editors, club officers and other amateurs initially not the least interested in packet radio similarly can be enticed to help. We haven't gotten the attention of many contesters, but that too will come in time. My point is that by division of labor and recruiting people with varied backgrounds, packet radio can get started with only a few "high-teckies."

Another possibility is to take advantage of the fact, as the saying goes, that necessity is the mother of invention. In those parts of the world where radio amateurs play a role in disaster communications, there already has been considerable interest expressed in packet radio -- and some significant use of the method, particularly in California. A marriage of technically inclined amateurs with those having a need or desire to improve Amateur Radio disaster communications can yield enormous benefits.

RADIO REGULATIONS

Radio amateurs in North America should consider themselves fortunate that their regulatory agencies (Department of Communications in Canada, and Federal Communications Commission in the United States) led the way by writing packet radio provisions in the rules. In the Canadian instance, in 1978, the DOC deliberately set out to foster packet radio, laid down specific rules for it, and created a Digital Amateur license class. In the U.S., the FCC took notice of the Canadian action and introduced both the ASCII code and packet radio into the regulations as the saving grace of an embattled inquiry already underway to control emissions by bandwidth rather than mode.

This is not to say that U.S. and Canadian radio rules are without problems with respect to packet radio. However, in both countries, the administrations took the initiative and have remained receptive to regulatory changes to encourage experimentation. If such receptivity to experimentation is lacking, the National Society should consider ways of working with regulatory officials to improve the climate. FCC officials have responded favorably to in-person briefings on, and demonstrations of, new technology. All regulatory officials having approving authority need to know what packet radio is, its potential, and its impact on others who share the spectrum. More fundamentally, they should understand how Amateur Radio experimentation benefits the general public and the communications industry.

Concerns

In addition to an appreciation of amateur experimentation and the benefits of packet radio, there are several concerns that need to be satisfied.

A major concern is that the regulatory agency may not be able to monitor packet-radio transmissions if and when they wish to. In dealing with this issue, we have pointed out that it is impossible for the government to monitor all Amateur Radio transmissions anyway because of propagation. Thus the propriety of Amateur Radio transmissions depends largely upon a trust that licensed amateurs will act responsibly and obey the law. The

Amateur Service takes pride in its ability to police itself. Confirmation of this bond of trust between amateurs and the government should do much to address the concerns that packet radio would be misused in the Amateur Service. Nevertheless, the regulatory agency needs the technical tools to monitor when it needs to. This can be satisfied by a packet-radio controller and a low-cost personal computer. The National Society can help by specifying exactly what is needed, facilitating procurement, and assisting in the initial installation if needed. Perhaps this can be obviated by getting one licensed enforcement official personally interested in packet radio to the degree of getting on the air. In certain countries, it may suffice to ensure that amateurs having the necessary qualifications and the full trust of their government have the ability to monitor packet transmissions.*

Even if it were technically possible to monitor every packet-radio transmission, there is the matter of volume. In the early stages of packet-radio development when volumes are low, it may be possible to closely monitor most or all transmissions within radio range. As volumes pick up, as they have in the U.S., human ability to physically read the transmissions is exceeded. It would be a full-time job for one person to scan traffic from one fully loaded packet-radio channel operating at 1200 bits per second. For an individual responsible for a digipeater, this poses a problem of how to prevent transmission of communications that are prohibited by the rules.

o An ultraconservative approach would be to have the digipeater trustee preview each and every packet prior to retransmission. That's obviously impractical and not considered necessary for voice repeaters; why burden a new technology with such a restriction?

o Another "cure" often suggested is to add a software "filter" that will screen out prohibited material. Some telephone-line bulletin-boards operators have gone so far as to develop a list of vulgar words. This has several pitfalls. There is a potential for embarrassment if the word list is revealed. Experience shows that it is impossible to think of every possible word or phrase that could be improper, especially if one is trying to rule out indecent language and other types of traffic (such as business messages). The "filter" ends up so all-encompassing that everyday communication between amateurs may not get through. Add to this the ability of individuals to outsmart the "filter" by using only words that are not on the list.

o The third approach is one based almost entirely on trust and peer pressure of the amateurs using the network. The trust aspect rests on the fact that amateurs are licensed, worked hard for it, and would not like to lose it by violating the rules. with a highly automated network, not too many amateurs will be monitoring each transmission, but those in range can do so.

Whether or not there is anyone monitoring, the packet still must be addressed to another amateur station; the addressee can be expected to advise the originator that a communication is improper. Operators of computer-based message systems (CBMSs), often called bulletin boards or mail boxes, can program their systems to prevent retransmission of messages not screened. Or, a more-liberal approach would be for the operator periodically to scan messages already stored, kill improper messages and advise the originator why the message was purged. This whole process can be backed up by a modest capability for volunteer monitors and governmental monitors to conduct spot checks and respond to complaints when they occur.

Signaling Rates and Spectrum Occupancy

About the only signaling rates in use at present are 300 and 1200 bauds. The lower speed is used on below 28 MHz; the higher one on VHF and UHF. If freedom to experiment were the only consideration, there should be no speed limit on transmission of data. But, of course, packet radio must share the spectrum with other users, so there should be some upper bound on the occupied bandwidth -- either by regulation or "gentlemen's agreement."

Unfortunately, it is not possible to equate data rate with bandwidth. Bandwidth is determined by several factors:

o data rate (in bits per second or bit/s)

o the modulation technique (e.g., FSR, BPSK)

o filtering and nonlinearities after filtering.

In the modulation systems used thus far, one bit is equal to one symbol transmitted (1 baud = 1 bit/s), thus bandwidth is proportional to data rate. However, as the spectrum becomes more occupied and

---

* For many years ARRL has had Official Observers. In 1984 an agreement was reached between the ARRL and FCC to establish an Amateur Auxiliary or Volunteer Monitoring Program. The agreement covers two classes of volunteer monitoring stations: 1) the station-level monitor or individual Official Observer, and 2) a handful of Regional Monitoring Stations.

modems become more sophisticated, there will be a trend toward so-called m-ary modulation systems. In such systems, it is possible to encode 2, 4, 8, or more bits into a single transmitted symbol by using various phase and amplitude combinations. It is important that radio regulations provide for m-ary modulation systems, for both experimentation and future growth of packet radio.

In the regulatory proceeding that set speed limits for digital communication, the FCC proposed defining speed in bit/s. Commenting amateurs pointed out that this would have locked out the use of m-ary modulation and would have been counter to spectrum conservation. As a result, where speed is mentioned in the rules, it is specified in bauds (symbols per second). Thus there is a direct relationship between the modulation rate in bauds and bandwidth. The FCC used both modulation rate and bandwidth, as summarized below:

Frequencies	ITA2 AMTOR ASCII	Any digital codes, only above 50 MHz
<28 MHz	300 Bd	Not authorized
28-50 MHz	1200 Bd	Not authorized
50-220 MHz	19,600 Bd*	20-kHz bandwidth
220-902 MHz	56,000 Bd	100-kHz bandwidth
>902 MHz	56,000 Bd	Any bandwidth within given amateur band

*19,200 is the standard rate.

The above modulation-rate and bandwidth limitations have served us well up to now and perhaps sometime in the future. But one can anticipate the need to press the limits upward over time. In a country starting with a "clean slate," it may be possible to incorporate more liberal provisions from the outset.

The 300-baud limitation at HF may be overrestrictive in that higher rates are possible. There are experimental modems developed by industry for the military that operate at 9600 bauds (serial not parallel signaling) in a single-sideband speech bandwidth (3 kHz). They use sophisticated "learning" techniques, require computer processing, and are beyond Amateur Radio pocketbooks for now but not necessarily forever. A speed of 1200 bauds would appear to be a reasonable upper limit for amateurs. Through the use of m-ary modulation techniques, actual data rates of 2400 or 4800 could be accomplished with learning and computer processing. If specified in bandwidth, with a spectrally conservative modulation technique and proper filtering a 1200-baud signal could be kept well within a 1500-Hz bandwidth.

On frequencies between 220 and 902 MHz, the modulation rate permitted by the FCC (56,000 bauds) is a standard rate for North America but not for the rest of the world, which follows CCITT guidelines for speeds. The CCITT number being recommended for the Integrated Service Digital Network (ISDN) is 64,000 bit/s. Perhaps U.S. amateurs should follow that standard rather than North American telephone practices.

Eventually it may be necessary to ask the FCC to raise the speedlimitabove 902 MHz to either "no limit" or something in the megabit-per-second range. Here again, there are some differences between North American and CCITT recommended speeds for "first-order" pulse-code modulation (PCM) networks, which are 1.544 and 2.048 Mbit/s, respectively.

Emissions

The new emission symbols adopted in the World Administrative Radio Conference (WARC-79) are more specific than those used for so many previous years. Furthermore, they do not correspond, one for one, with the old ones. In addition to describing the signal as it appears on the air, the new symbols also are specific as to how the signal is generated. This makes it difficult to translate old symbols into new ones.

For packet radio, it is desirable to have a very broad description of permissible types of modulation in the rules. It might be possible simply to specify something like "data transmission, telemetry and telecommand by any amplitude modulation, angle modulation, or a combination thereof, using bandwidths in keeping with good engineering practice." Amplitude modulation and angle modulation cover everything except pulse modulation. For reasons probably needing reexamination today, pulse modulation is not permitted on the lower-frequency bands. Pulse modulation would be valuable for future higher-speed packet-radio applications.

The other alternative is to specify every possible modulation scheme by emission symbol. This becomes cumbersome. For example, the old symbol, F1 could be translated to F1D, but that covers only direct frequency shift of the main carrier. F1D does not include phase-shift keying (PSK), which would be G1D. To make things more complex, either frequency or phase shift of a subcarrier modulating a single-sideband transmitter would make the emission symbol J2D. Further, it is possible to suppress the sideband of a high-speed data transmission as commonly done for speech: that would be J1A. There could be other specific emission designators if two or more channels are multi-

plexed or if pulse modulation is used, both of which could occur at megabit-per-second speeds.

It appears that the better approach to emissions, particularly for packet radio, is to ask the administration to give amateurs only broad guidelines that will not stifle experimentation. That would also ensure that international packet-radio communications will not be hampered by incompatibility caused by overspecificity.

Station Identification

At one time, digital transmissions had to be identified by Morse code under FCC rules. It was liberalized to permit identification in any of the specified digital codes (ITA2, AMTGR and ASCII). Using the AX.25 link-layer protocol, the call signs of the addressed station and sending station are sent at the beginning of each packet transmission, in ASCII. This meets FCC identification requirements and lets any monitors know who is transmitting and who is intended to receive the packet. Where digipeaters are used, the AX.25 address field is extended to include the call signs of each digipeater along the way.

It appears that the AX.25 addressing arrangement meets at least the spirit of identification requirements anywhere. However, the regulations of some administrations may require amendment or at least reinterpretation. For example, a national licensing authority may require that the call signs of the addressed and sending stations be transmitted at the beginning and the end of each transmission. It may be possible to successfully argue that the beginning and end of a packet lasting only one second are so close together that only one identification is needed per packet.

Conformity to Widely Recognized Standards

When AMTOR was new, FCC acceptance of this mode was easy, in part, because of the existence of CCIR Recommendation 476-2 specifying this mode for international maritime use. The FCC authorized AMTOR by simply incorporating by reference Rec. 476-2, and later 476-3, in the rules. It may be generalized that following industry and international standards strengthens a case in petitioning regulatory authorities for rules changes in the Amateur Service.

If rules changes are needed to permit packet radio, it can be stated that packet-switching techniques are now in widespread use throughout the world in other communications services. The AX.25 link-layer protocol follows a number of international standards, principally:

0 International Organization for Standardization (ISO) standard IS0 3309, Data communication--High-level data link control link procedures--Frame structure.

o International Telegraph and Telephone Consultative Committee (CCITT) Recommendation X.25, Interface between data terminal equipment and data circuit-terminating equipment for terminals operating in the packet mode on public data networks.

In development of packet-radio standards and practices, the ARRL approach has been to follow international (as distinguished from national) standards to the degree feasible. Amateurs in the U.S. are using modems that conform to North American Bell Telephone standards (not CCITT) for packet radio. This practice was brought about by the ready availability of Bell Telephone modems at surplus prices. Fortunately, Bell and CCITT modem incompatibilities are somewhat moot when used via Amateur Radio. On HF, for example, Bell 103 and CCITT V.21 can communicate through SSB transceivers because the frequency shift in both cases is 200 Hz; the difference in tones is easily compensated for by tuning of the transceiver. On VHF and UHF, there is little radio contact between North America and other continents except via satellite. The ARRL has not taken a position on packet-radio modems standards to date. However, any future recommendations will be developed with due consideration to international standards. The existence of integrated circuits capable of multiple Bell and CCITT modem protocols helps to diffuse modem standards as a serious issue. A modem using the AM7910 chip is shown in the current <u>ARRL Handbook</u>.

Third Party Traffic,

Perhaps the most sensitive issue is third-party traffic. Packet radio is technically suited to handle third-party traffic where permitted and may, in time, supplant manual transmission methods. Rules governing third-party traffic are quite liberal in the United States and certain other countries. There are certain restrictions to prohibit competition with common carriers and other commercial radio services. Yet, we are aware that other philosophies govern third-party traffic rules in other countries; many outlaw it entirely, while others have exceptions only for declared emergencies. In some cases, the regulatory language prohibiting third-party traffic was so broad as to rule out repeaters.

As the packet-radio network grows, there will be a "technical imperative" to have message traffic relayed from one country to another possibly through intermediary countries. If the third-party-traffic rules are nonuniform, routing requirements could become chaotic for international messages. Or, unfortunate-

ly, amateurs may simply choose to ignore provisions in the rules that seem inconvenient. Thus, the technical imperative takes over: It is **possible;** therefore do **it!**

It may be appropriate for the **IARU to** take the lead in developing **"model** regulations" pertaining to third-party traffic. If there is broad international agreement on a workable model based on a set of there is a good chance of favorable consideration by licensing authorities.

CONCLUSIONS

Packet radio is here now and is growing at a substantial rate. There is a sufficient technical base for its development throughout Region 3. National Societies can encourage its orderly growth by providing accurate and timely **information,** and by supporting their centers of excellence, as detailed above. Liaison with regulatory authorities is also a special role of National Societies; goals should be to a) assure that officials have a proper appreciation of this new technology and b) convince the authorities of the need to modernize regulations to accommodate new modes such as packet radio.

REFERENCES

ARRL, **ARRL Handbook for the Radio Amateur,** 1985 and 1986 eds.

ARRL, Gateway -- The **ARRLPacket-Radio** Newsletter, fortnightly.

ARRL, Proc., First through **Fourth ARRL Amateur** Radio Computer **Networking Conference,** 1981, **1983, 1984, 1985.**

FOX, **AX.25 Amateur Packet-Radio Link-Layer** protocol, **ARRL,** 1985.

Karn, Price and Diersing, "Packet Radio in the Amateur Service," **Selected Areas in Communications,** IEEE Communications Society, May 1985.

Price, **"What's** All This Racket about Packet?," **QST, July** 1985.

Price, **"A** Closer Look at Packet Radio," **QST,** August 1985.

Gateway Premier Issue

The ARRL Packet-Radio Newsletter August 14, 1984

WELCOME

Welcome to Gateway, the ARRL packet-radio newsletter. Some of you reading this are deeply involved in amateur packet radio, some of you are just getting started on this exciting new mode, and some of you aren't quite sure what "packet radio" is. We hope that Gateway will have something to offer each of you.

WHAT IS A "GATEWAY?"

A gateway is a station that link6 two communication6 networks. In amateur packet radio, gateway stations are being used communicate between VHF and HF networks, and between VHF network6 and satellite channels. Eventually, users of local VHF networks will use gateways to connect to an internat ional packet-radio network. We have called this newsletter Gateway because we hope that it will, like a gateway station, facilitate communications between amateurs interested in packet radio.

Gateway will not be a technical newsletter; there are already several 'fine packet-radio newsletters covering technical issues. This will be a "news" newsletter. At ARRL Headquarters we have many sources of news not all of which are available to each of you. This newsletter will bring together notes from these sources. Overseas and domestic packet-radio club newsletters, the FCC, the IARU, on-line conferences and on-the-air bulletin boards will contribute. You may see items that you've seen elsewhere, but you should also see things that are new and interesting.

Some of you are receiving this packet-radio newsletter and have never even considered what packet radio can do for you, or what fun you could have on packet. We hope to provide you with an overview of the state of amateur packet radio, explaining what is being done and what can be done on this new mode.

It seems as though every packet-radio club is undertaking some project that will advance amateur packet radio. To make these projects fruitful, we must make the most of the limited resources available to amateurs. By telling a large audience about various packet-radio development efforts, Gateway should help organizers direct their efforts, and help volunteers find the groups that need them.

Perhaps when there is a worldwide amateur packet-radio network there will be no need for packet-radio newsletters. Until then, we hope that Gateway informs and interests you.

PACKET METEOR SCATTER

Last weekend's Persieds meteor shower provided a good opportunity to experiment with packet-radio meteor-scatter operation. Rich Zwirko, KlHTV, set forth some experimental guidelines, and stations throughout the U.S. attempted MS QSOs.

On August 1, well before the peak of the Persiecis, WØRPK, the station of the Central Iowa Technical Society, held schedules with Bob Carpenter, W3OTC, in Maryland. The skeds were on 6 meters. Bob received about 2% of the packet6 sent from Iowa. Four nights later, again on 6 meters, W3OTC and WØRPK had what is believed to be the first amateur packet-radio meteor-scatter QSO.

As the Persieds approached, stations tried their luck on the Z-meter band. Stations in the East included KlHTV; Vern Riportella, WA2LQQ; Tom Clark, W3IWI; Hank Oredson, WØRLI; and Mark Wilson, AA2Z. In the West were Ralph Wallio, W0RPK; Bob McCaffrey, KØCY; Ron Dunbar, WØPN; Mike McQuiston, WAØWYW; Bob Schiers, NØAN; and Terry Van Benschoten, WØVB. Several of these stations copied beacons and connect requests via MS. On the morning of August 12, during the peak of the shower, WØRPK and KlHTV completed the first packet MS QSO on 2 meters. Congratulations are in order for all stations involved in these tests, and I hope that I haven't left anyone out.

These tests were performed at 1200 bauds, using AFSK FM. While it was necessary to use this mode in order to include as many stations a6 possible in the experiments, a performance sacrifice was made. We should organize further tests at higher transmission rates with more efficient modulation techniques.

ARRL APPOINTED IARU PACKET CENTER

The ARRL has been appointed IARU packet radio "information clearing house." The following is exerpted from the minutes of the IARU meeting in late July: "ARRL is nominated as the international clearing house of information relating to packet radio on behalf of the IARU, with a view to encouraging common standards and regulations."

This points the way toward international understanding and acceptance of packet radio standards generated in the United States and Canada. Several European amateurs have hesitated to get involved in packet radio because they were not sure which standards would "catch on." The appointment should help to alleviate this confusion abroad as to what is really happening in North American packet radio.

PACKET RADIO ON NETWORK TV

On July 16 at 0745 PDT, a packet from N6ECT was heard throughout the nation on the CBS Morning News show. Curtis Spangler, N6ECT, was being interviewed by CBS for a piece on the Haight/Ashbury and the film crew focused on his CRT. As luck would have it, Curtis was transmitting packet6 through the KA6M digipeater. The audio from one of the packet6 came through loud and clear, and the frame was heard in all 50 states and throughout half the world! Via KA6M.

220 MHz and PACKET RADIO

The 220-MHz band is crucial to packet radio for a couple of reasons. Because it is the lowest frequency band on which we can exceed 19,600 bauds, it is going to be used for the initial high-speed intercity linking. It is not being used for any satellite uplinks or downlinks, and 60 it is essential for full-duplex teleport stations. With these considerations in mind, we note the following:

The Tri-State Amateur Repeater Council has coordinated a 100-kHz channel from 220.5 MHz to 220.6 MHz for wideband digital communications. This council coordinate6 VHF frequencies in northern New Jersey, southwestern New York (including Long Island) and Connecticut. This paves the way for EASTNET linking to begin as soon as hardware and software are ready.

On the negative side are two petitions for rule making which threaten the 220-MHz amateur band. The first is RM-4829 from the Land Mobile Communication6 Council. This petition calls for the FCC "to explore the potential use of vacant spectrum in the UHF TV bands, spectrum allocated for 'Fedral Government use, or assignments from the band 220-225 MHz to satisfy the requirements of land mobile users." This may sound worse than it is, since the petition goes on to say "Because of the limited number of channel6 that the 220-225 MHz band will provide, however, it is not anticipated that this spectrum can meet the immediate requirements of land mobile licensees."

Another petition (RM 4831), from a manufacturer of amplitude compandored sideband equipment) explicitly requests reallocation of the band 216-225 MHz. This petition poses a serious threat to the 220-MHz amateur band.

Packet radio needs the 220-MHz band. Be sure to read these petitions and send your comments to the FCC. The comment procedure was outlined in QST, March, 1982. The comment deadline for these two petitions is August 29, 1984.

WHAT'S BREWING AT TAPR?

This piece came from Harold Price's answer to the question "Is TAPR up to something?" Harold is part of the TAPR software design team, and he made these comments while he was "Member of the Month" on Compuserve's HAMNET.

"The following views are mine alone, and do not necessarily refect those of TAPR, AMSAT, VITA, LAPG the staff, management, or janitorial departments.

"The TAPR folk6 are indeed up to something. We have the TAPR Pascal code running happily under a simulated environment again. The software, with only one change, runs under TURBO Pascal on the Pronto-16. This will vastly speed up development, which has slowed down as of late.

"The plan is to come up with version 4.0 of the TAPR TNC software which will allow testing of both datagram and virtual connection protocols. I think the level two wars are over. With. 1300 TNCs in the field from 6 "manufactuers" all running the same level two, anyone proposing a switch now is just rocking the boat. The few proposals I've seen for different level twos offer no concrete advantage6 over what we've got now anyway. Besides, level two is boring (now that we have one that works). The real fun is level three.

"For the newcomer, level two refer6 to a point to point protocol, linking one TNC to another with no TNCs in the middle. There is currently a necessary kludge in AX.25 called digipeating which is a very demented level three feature. Digipeating allows two TNCs to be connected using a third as a relay. Without this simple addition to AX.25, packet may not have taken off as it did, since digipeating allows many more users to reach each other. If you haven't got a wide-coverage duplex repeater (or even if you have), digipeating is your best bet for now,

"Anyway, level two is point to point, with level two+ in current style, multihop dumb repeating. The + in level two will die a happy death when we get level three up and running. Level three link6 two end points thru multiple intermediate TNCs. The linking is done in an intelligent manner. ACKing is node to node rather than end to end. In level two digipeating, each intermediate point simply regenerates the packet. It does not ACK it. The final end point ACKs it, and the ACK is blindly repeated back to the starting point. If any repeat of the packet, or the ACK, is stepped on or dropped, the packet must start over from the beginning.

"In level three, an ACK can occur at each step of the way. Thus, a packet may only have to be re-sent between relay points five and six, rather than starting again at point one. So why don't we get on with it, you might ask?

"There are many problems involved in design and implementation of a level three network -- flow control, network blocking, routing, on and on. What is TAPR doing?

1) A node in a level three network will want to be connected to more than one other node. We will allow the TAPR TNC to maintain multiple level two connects. This has several implications. First, you can carry on two or more concurrent conversat ions. Not so good for rag chewing maybe, but great for emergency communications. Imagine a TNC in the local disaster center. Currently you can carry on a conversation with only one other TNC, with limited possibility of a priority break-in from another TNC. With all outlying TNCs connected to the central node at the same time you get closer to what you want, high reliability connections with each of the field guys at the same time.

An Introduction to Packet Radio

Jon Bloom, KE3Z
ARRL Laboratory Engineer

Perhaps you have heard or read of packet radio but are unsure just what this new mode is. Does the idea of error-free communication at high speed interest you? How would you like to use a remote computer to get propagation charts, satellite tracking data or leave messages for your friends? Or maybe just play a computer game? All of this, and more, is possible via packet radio.

No More "Hits"!

Packet radio is a digital communications mode. Don't be scared off by this impressive sounding phrase; old-fashioned RTTY is a digital mode too. So is CW in the most literal sense! A digital mode is simply one which is transmitted by sending a signal with only two states: on and off for CW, mark frequency and space frequency for RTTY. In fact, packet radio serves much the same purpose as RTTY but packet has a number of improvements that make it more versatile and useful. One of the major failings of RTTY is its susceptibility to errors. In other words, as signals fade or QRM appears the "print" suffers errors. When this occurs in a RTTY QSO it is impossible to determine what was actually supposed to be printed. This is probably the major area in which packet radio improves upon RTTY. When an error occurs in the received packet signal, it is detected, and the information is automatically retransmitted. In fact, retransmission of the information will keep occurring until the receiving station transmits an acknowledgement, or ACK, to the transmitter. Since the error-detection technique is very sophisticated, there is almost no chance of accepting erroneous information. This automatic repeat request is known as ARQ.

To make the ARQ system work, the transmitter must pause from time to time to allow the receiver to send its ACK. In packet radio, the transmitter will send a burst of information (a packet) to which the receiver will reply with an ACK signal if the packet was received without errors. If the transmitter does not get an ACK back, it waits a short time and then retransmits the packet. Unlike other digital modes such as RTTY and AMTOR, the packet station waits until it has a number of characters to send before transmitting. For "chat" type operation, the packet station will generally transmit the information after a complete line has been typed or after a specified number of characters has been entered. The station then activates the transmitter, sends the information, goes back to receive mode and waits for the ACK. If the ACK is received right away, the transmitter will remain off until the next line is ready to be sent. So for much of the time neither of the stations are transmitting at all! It is this characteristic of packet radio which allows another benefit of the mode: channel sharing. It is not at all uncommon for two or three (or more> QSOs to be taking place on the same frequency at the same time! Of course, occasionally two stations will transmit at the same time and clobber one another, but because of the ARQ function they will each try again, and the information will get through. Since the VHF packet signals are sent at 1200 bauds (more than 20 times as fast as normal Baudot RTTY) it doesn't take too long to get the information through, either.

Protocols – The Bedrock of Digital Communications

To make packet radio work, all of the stations need to be using the same <u>protocol</u>. Again, don't let this term make you nervous. Hams have been using protocols for many years. A protocol is simply a defined way of doing things. The standard amateur message format is one example of a protocol. It exists so that all traffic handlers will know exactly what information is required to refer to a given message and so that the receiving operator will know what to expect next as the message is being received. Another protocol is the RTTY character frame. It has a particular make-up consisting of a start bit of specified frequency and length, five information bits in the proper sequence and a stop bit. Imagine the confusion if each station sent a different length start bit or sent the information bits in any convenient order! In packet radio, there are actually several protocols in use. Each of these protocols supports the operation of the protocol above it. We say "above" because of the way we look at how the packet protocols support one another. The model we use is the International Organization for Standardization (OSI) network reference model. This model shows seven layers of protocol.

In packet radio, the first protocol layer defines the way in which the information from the higher layers is sent as a sequence of mark and space frequencies (or amplitudes or phases) by the transmitter. On VHF, packet radio stations currently use Bell 202 modems to transmit and receive AFSK signals. Standards for HF use are still being developed.

The layer 2 protocol specifies how the information is converted to a sequence of bits and what information is sent by the two stations to insure that the packets wili be printed in proper sequence and without errors. Layer 2 consists of *two* protocols. The first of these is an international standard known as High Level Data Link Control (HDLC). (Confused? Don't worry. You don't have to understand all of this background information to operate packet. It's just nice to have *some* idea what the packet experimenter types are talking about!) The second layer 2 protocol in use by amateurs is called AX.25 and is a variation of the commercially used X.25 standard layer 2 protocol. The major change which was made by the amateurs is the inclusion of call signs in each and every frame of the transmissions. As we mentioned before, more than one QSO can occur on a given frequency at the same time. How is a receiving station to know which of the transmissions are part of his own QSO? This is accomplished by addressing, each frame with the call sign of the receiving station. The call sign of the transmitting station is included as well. Aside from the need of this addressing to make the protocol work, it makes it possible for stations monitoring the frequency to see who is transmitting. It also constitutes a legal station ID for FCC purposes, This eliminates the need for the operator to explicity identify his station.

Packet Radio Today

Most of the packet-radio activity today is carried out using standard 2-meter FM radios. Along with the FM rig, the packet station consists of computer or video terminal and a terminal node controller (TNC). The TNC is a small microcomputer which executes the packet-radio protocols. It sends the information for display to the terminal or computer and gets the information to send from the same device. Several manufacturers now produce TNC kits and assembled units.

A repeater is commonly used to extend the range of VHF FM stations. Although a standard voice repeater may be used to repeat packet signals, there is a simpler way. The digipeater is a packet-only repeater. Along with the call sign of the transmitting and receiving stations, a packet frame may contain one or more digipeater call signs. When a digipeater receives an error-free packet frame it checks for the presence of its own call sign. If the call sign is found, the digipeater retransmits the frame. Since the digipeater does not operate in full-duplex mode, it requires only a single frequency. Also, it does not need any duplexer or sophisticated control circuitry as does a voice repeater. A digipeater need only consist of a standard 2-meter rig, a TNC and a good VHF location! With more than one digipeater address in the frame, the packet may be made to hop from one repeater to the next. Large distances may be traversed in this way.

The Future of Packet Radio

A packet network is formed when a number of stations can communicate with one another through a common set of communication links. A rudimentary network is formed when a number of stations are connected directly or through digipeaters. A more effective network would have the ability to intelligently repeat the frames along a route between the sending station and the destination station regardless of the distance between the endpoints. It is this facility that is the next major effort to be undertaken in packet radio. To accomplish this, a level 3 (network) protocol must be specified and implemented. The expectation is that level 3 should become a reality in 1985. As more packet stations come on the air, QSOs between far distant stations will be possible. Imagine using a 2-meter hand-held radio to communicate with stations across the country! Soon it will be possible.

Chewing the rag is close to the ham's heart, but that type of operation only scratches the surface of the potential of packet radio. Computerized server stations are providing bulletin-board and remotely operated computer services now. More esoteric servers will be built in the coming years. Such new possibilities as digitized voice and video have barely been explored to date, leaving plenty to do for both experimenters and users. The OSCAR satellites are being used for packet radio now. With the packet-radio satellite (PACSAT) scheduled for launch in 1986, space communications promises to play a big part in packet. The high frequencies have not been neglected either. Experimental packet transmissions on 10 and 28 MHz have proven the viability of packet as an HF mode. Improved techniques for HF use are being developed now. Expect *to see* packet on HF in a big way within a short time.

Amateur Radio is not being left behind in this age of the home computer revolution. Packet radio allows the amateur to combine his or her interest in the radio art with the high-tech world of computers. With the communication potential of amateur radio added to the information processing of the computer, the coming years should show a dramatic increase in the capability of the typical ham stat ion. Get in on the ground floor now!

Bibliography:

ARRL Amateur Radio Computer Networking Conference (Newington: ARRL, 1981).

Second ARRL Amateur Radio Computer Networking Conference (Newington: ARRL, 1983).

Third ARRL Amateur Radio Computer Networking Conference (Newington: ARRL, 1984).

D. Borden and P. Rinaldo, "The Making of an Amateur Packet-Radio Network," QST, October 1981

I. Hodgson, "An Introduction to Packet Radio," Ham Radio, June 1979

L. Johnson, "Join the Packet Radio Revolution," 73 Magazine, September 1983

L. Johnson, "Join the Packet Radio Revolution," 73 Magazine, October 1983

M. Morrison and D. Morrison, "Amateur Packet Radio: Part 1," Ham Radio, July 1983

M. Morrison and D. Morrison and L. Johnson, "Amateur Packet Radio: Part 2," Ham Radio, August 1983

"Next, and even better, you can automatically route one connection **stream** to another. Maybe an example is called for. The syntax below is probably not what we'll end up with, but the idea is:

```
MYCALL NK6K
[1]CONNECT WB6YMH
     [1] CONNECTED TO WB6YMH
[2]CONNECT WA6JPR
     [2] CONNECTED TO WA6JPR
ROUTE [1] TO [2]
```

"Your TNC is now a network node. Anything that comes **in** from stream [1] gets ACKed at level two. The data from the packet gets routed to stream [2] where it gets sent out and saved until an ACK **comes** in on stream [2]. The reverse is also true, incoming from [2] goes to [1].

"Now, wouldn't it be great if you could cause the other guys board to make a connection? If I could tell WA6JPR to make a level two connection to WB5EKU? Then what we have is the level three function, endpoints linked thru multiple intermediate points. A lot of things are missing, but this simple mechanism allows testing of level three concepts without a lot of hassle on the users part. We will also design an interface (based on asyncronous LABP) between the TNC and its attached computer to allow the computer full control over the link process. This permits the use of the TAPR TNC as a level two black box, with level three functions done in your host.

"Do I expect everyone to run verson 4? Well, why not? Version 3.1 can still be used point to point, and thru 4.0 gateways get full access to network. But just as everyone having the capability of being a digipeater added to the swift growth of packet, so will the ability of each TNC to be a level three node.

"But I ramble. Not only does a network node need to support multiple level two connections, it might also need to support connections on multiple RF channels. Let's assume a 1200-baud link on 2 meters and a 2400-baud link on 220, feeding a 9600 baud link on 440. The hardware arm of TAPR is designing a fancy multiport hardware controller. Several designs have been proposed, including a motherboard with slots for plugging in a number of channel-controller cards. Each channel-controller card is a mini TNC, handling all channel-access and level two functions. The mother board passes data between channels and handles level three and higher functions.

"We don't expect everyone to have one of these TNC-LINKS (say "tink link"). But they will make great mountain-top controllers, especially when used with the PACSAT 9600 whiz-bang modem, which has been described elsewhere.

"How long do I see TAPR building kits? There are two answers: "As long as there is a demand" or "Until we can't stand the sight of them anymore." I haven't been pressed into the chain **gang** of kit packaging, but isn't much fun, especially when serial number 1000 has long since gone out the door. The kits are TAPR's only source of income. An extremely small amount of each $240.00 goes into our fund for future development. I have forgotten how much. A number that does come to mind is the cost of the cabinet kit. Your cost, $69.00, our cost, $67.00. Our original goal was to make packet available to a large number of people at reasonable cost, delivering as full a function device as we could. It is possible to deliver less function for less cost. It is possible to deliver **the** same function assembled, tested, and warranted, for a larger cost. There are several market niches out there, and we will continue to ship as long as 1) there is a niche for us and 2) we're having a good **time**.

"Are we having fun yet? You bet!

"One final note. A for-profit company would be crazy to discuss future products like this before the product is ready to ship. But we're a nonprofit R&D company, trying to make packet the mode of the future. And remember, you saw it here first." Via HAHNET.

<u>Club Listing</u>

Here is a list of clubs active in Amateur packet radio .

Amateur Radio Research and Development Corp. (AMRAD)
P.O. Drawer 6148
McLean, VA 2210606148

Amateur Radio Satellite Corp. (AMSAT)
P.O. Box 27
Washington, DC 20044

Chicago Area Packet Radio Association (CAPRA)
P.O. Box 8251
Rolling Meadows, IL 60008

Central Iowa Technical Society (CITS)
c/o Ralph Wallio, WØRPK
Rural Route 4
Indianola, IA 50125

Florida Amateur Digital Communications Assn. (FADCA)
c/o Ted Huff, K4NTA
1829 N. W. Pinetree Way
Stuart, FL 33494

Los Angeles Area Packet Group (LAPG)
c/o Harold Price, NK6K
1211 Ford Ave.
Redondo Beach, CA 90278

Minnesota Amateur Packet Radio (MAPR)
c/o Philip S. Plumbo, NØDFT
1128 Dayton Avenue
St. Paul, MN 55104

New England Packet Radio Assn. (NEPRA)
P.O. Box 15
Bedford, MA 01730

Rocky Mountain Packet Amateur Radio Assn. (RMPAR)
c/o Andy Freeborn, NØCCZ
52222 Borrego Drive
Colorado Springs, CO 80918

San Diego Packet Group (SDPG)
c/o Mike Brock, WB6HHV
10230 Mayor Circle
San Deigo, CA 92126

Tucson Amateur Packet Radio Corp. (TAPR)
F.O. Box 22888
Tucson, AZ 85734

Vancouver Amateur Digital Communications Group
(VADCG)
9531 Odin Road
Richmond, BC V6X 1E1
CANADA

Remember to include an s.a.s.e when writing to any of these clubs. Club money spent on postage is money that can't go into packet-radio development.

We also know of the following individual who would like to form a packet-radio club:

Doug Baker, K4CLE
8724 Cumbernauld Circle
Germantown, TN 38138

Voice Nets Concerning Packet Radio

The following voice nets are devoted to discussion of packet radio.

Day	Time	Frequency	Coverage	Sponsor.
Mon	2000 PDT	145.36	Los Angeles	LAPG
Tue	2100 PDT	144.76	San Diego	SDPG
Tue	2200 EDT	3.850	Eastern U.S.	'AMSAT
Thu	2000 EDT	147.12172	Boston area	NEPRA
Sun	1900 UTC	14.235	National	TAPR
Sun	2100 UTC	7,158	National	TAPR
Sun	0800 EDT	3.958	Regional	FADCA
Sun	1900 EDT	147.165	Tampa area	FADCA
Sun	2100 EDT	147.285	Orlando area	FADCA
Sun	1500 CDT	7.158	Central U.S.	CITS

Gateway: The ARRL Packet Radio Newsletter is published by the

American Radio Relay League
225 Main Street
Newington, CT 06111

Larry E. Price, W4RA
President

David Sumner, K1ZZ
General Manager

Jeff W. Ward, K8KA
Editor

Subscriptions are available to ARRL members and nonmembers, at the following rates:

For 25 issues of Gateway:
- In the U.S.
  - ARRL Member $6 .OO
  - Nonmember $9.00
- In U.S., Canada, and Mexico by First Class Mail
  - ARRL Member $11.00
  - Nonmember $14.00
- Elsewhere by Airmail
  - ARRL Member $14.00
  - Nonmember $17.00.

Prices are subject to change without notice. Remittance must be in U.S. funds, and checks must be drawn on a bank in the U.S.

Gateway: The ARRL
Packet Radio Newsletter
American Radio Relay League
225 Main St.
Newington, CT 06111

Nonprofit Organization
U.S. Postage
PAID
Hartford, Conn.
Permit No. 2929

Gateway

Vol. 1, No. 2
Aug. 28, 1984

The ARRL Packet-Radio Newsletter

GATEWAY SUBSCRIPTIONS

If you have received the first two issues of Gateway for free, now is the time to subscribe. We have to recover the cost of producing, printing, and mailing the newsletter. Subscription information can be found on the back page.

GATEWAY INPUT

Is there packet activity in your area which you think would be of interest to the readers of Gateway? If there is, send it along to ARRL Hq, marked "Attention: Gateway Editor." This newsletter was started to distribute information about packet radio throughout the world. Send us a summary of activity on your packet network or tell us of any packet-radio related R+D work going on in your area.

HF PACKET BULLETIN BOARD

You can now connect to the WØRLI packet bulletin-board system (PBBS) on twenty meters. The PBBS is located near Boston, MA, and is run by Hank Oredson, WØRLI. Hank has had the PBBS on 145.Ø1 MHz for several months, and it has become one of the major "servers" for the large Boston-area packet network.

The WØRLI PBBS software, written by Hank in Z8Ø assembly language, runs on a Xerox 82Ø with a TAPR THC. The PBBS provides automatic time/date stamps for messages, automatically delete6 inactive messages, and will even try to send you a beacon if there is mail waiting for you. Now, with the addition of a second TAPR TNC, the bulletin board is available on both HF and VHF.

On twenty meters you will find the PBBS as near 14.080 MHz as band occupation permits. Use 3ØØ bauds, and 2ØØ-Hz shift. (The same shift and speed used by 1Ø-MHz packet stations.) When you connect to WØRLI, the PBBS will send you operating instructions. Via WØRLI.

NTS TRAFFIC VIA PACKET

WØRLI's PBBS is also being used to originate NTS traffic. Messages are stored in standard NTS format, with the preamble information on the first line of the message. The PBBS is checked each evening and messages are then delivered to the appropriate NTS net.

Packet radio is a valuable tool for traffic handling. Of course, the national packet network that we are working toward will be a perfect traffic-handling tool, but what we already have can and should be connected to NTS. The existing local networks can be used for message origination and delivery. Two packet stations on 1Ø MHz or OSCAR-1Ø could provide TCC service. These operations could begin today if NTS officials contacted packeteers, or packeteers contacted NTS officials. Let's get started!
Via The NEPRA Packetear.

PACKET RADIO AND ARES

Among packeteers, it is "common knowledge" that packet radio would be a good mode for the high-volume, error-free communications needed during emergencies. Unfortunately, this knowledge has yet to reach the leader6 and operators in the Amateur Radio Emergency Service (ARES). To remedy this, a new chapter has been added to the ARRL's Emergency Coordinator's Handbook.

In July, Mike Riley, WF4R, editor of the EC Handbook, met with Terry Fox, WB4JFI, of AMRAD. The two discussed ways in which packet radio could aid the ARES. The new ten-page "Packet Radio" chapter in the EC Handbook is the result of that discussion.

The EC Handbook is sent free-of-charge to all Emergency Coordinators. The latest edition, vith the packet-radio chapter, will be available early in September. At that time, packet-radio enthusiasts should be prepared to discus6 emergency operations with their ECs.

HIGH-SPEED PACKETS

On August 23, Curtis Spangler, N6ECT, and Mike Flynn, W2FRT, exchanged packets at 96ØØ bauds using quadrature amplitude modulation (QAM) techniques. Both stations were using personal computers, 96ØØ-bit/s modems, homemade radio/modem interfaces, and 44Ø-MHz radios. Special software, vritten in Turbo Pascal, drove the synchronous data link controller (SDLC) cards in the computers. Over the five-mile path betveen the stations, there were no errors using 1Ø watts, and 6Ø% to 7Ø% throughput at one watt. Via KA6M

TELEPORT STA

On August 6, the ARRL and AMSAT filed a request for special temporary authority for the operation of teleport stations. The following comes from the text of the STA:

"The operations of the stations operating under the STA will involve communications vith terrestrial Amateur Radio Stations and Amateur Radio stations in space operation. They will function primarily as intermediary stations betveea terrestrial and space Amateur Radio stations. We hereby designate these intermediary stations 'teleports.'

"The purpose of a teleport is to relay digital messages automatically between terrestrial Amateur Radio stations and amateur stations in space operation (amateur satellites). This need to do so is twofold:

"(1) to obviate the need for every Amateur Radio station having a digital communications capability to also have an earth station capability in order to communicate vith amateur satellites.

"(2) to provide a measure of traffic-flow control for the digital channel(a) on the satellite(s)...

"There are three primary objectives of this request for STA:

"(1) to determine experimentally what equipment and techniques are required to provide near-real-time relays between two or more terrestrial rtatioas using local teleports and an amateur satellite.

"(2) to determine experimentally what is required to provide reliable store-and-forward communications wherein the teleport station buffer stores the messages between the hours that the satellite and terrestrial links are not available at the same time.

"(3) to gather the necessary information for permanent rule change to permit teleport operation ."

The request for STA then goes on to list the frequencies of operation, and other administrative details. The folloving stations will be authorized to operate teleports when the STA is approved: John Biro, K1KSY; Tom Clark, W3IWI; Den Connors, KD2S; Bob Diersing, N5AHD; John DuBois, W1HDX; David Engle, KE6ZE; Gary Garr iot t, WA9FMQ; Sumner Hansen, WB6YMH; Lyle Johnson, WA7GXD; Phil Karn, KA9Q; Bob McCaffrey, KØCY; Harold Price, NK6K; Bill Reed, WDØETZ; Hank Magnuski, KA6M; Vem Riportella. WA2LQQ; Jose Sancho-Dominguez, WB5YFU; Bob Stric' .in, N5BRG; ARRL club station, W1AW; and AMRAD clu. station, WD4IWG. Building a teleport is a demanding task, and no one will have a tefeport immediately. Please do not bother these operators by asking when their. teleport will be on the air.

OSCAR-1Ø and UoSAT-OSCAR-11 are the satellites that are likely to be used by the authorized stations. OSCAR-l& vith its long access times and great coverage, will be used to test the near-real-time links. UoSAT-OSCAR-11, if it is made available at all, will be used for store-and-forvard teleports.

Watch Gateway for further nevs of the STA and the stations involved.

PACKETEERS ARE EVERYWHERE

meets the eye! This vas brought home to me on a recent weekend vhile my vife Linda, KA1ZD, daughter Der.yn, and I were vacationing in Nev Hampshire. As we were vandering around a computer tent sale, someone spotted Linda's HT and said, "I bet you're a ham." He turned out to be Dave McLanahan, WA1FHB, and ve had a long and pleasant' conversation about all sorts of things. Eventually I introduced the subject of packet radio, and asked innocently if Dave vas familiar with the mode. He was not only familiar with packet, he was on it! Living in a bit of an RF hole in the vilds of southwestern New Hampshire, Dave goes hilltopping vith a ZX81, a S-inch TV monitor, and a GLB TNC to connect into the Nashua, Nev Hampshire area. With EASTNET growing as it is, he should soon be able to packet without leaving his home. Via Dave Sumner, K1ZZ.

[Dave is General Hanager of the ARRL -- Ed.]

PACKET RADIO IN AUSTRALIA

There are at least two active packet clubs in Australia, one in Melbourne and one in Sydney. The newly-formed Melbourne Packet Radio Group has four members: John Smelstorius, VK3ZVR; Ian Clark, VK3YRR; Peter Jetson, VK3ZMB; and David Furst, VK3YDF. These four have started a local net using the VADCG protocol. They will be attempting to link to the larger group in Sydney.

The Sydney Amateur Digital Communications Group (SADCG) has an active net of about twenty stations with a digipeater and a packet bulletin-board sys tern.

Perhaps someone will eet up an HF or satellite gateway linking the U.S. and Australia.
Via Amateur Radio.

THE PACKET ADAPTIVE MODEM

If amateurs are to construct a long-distance packet network in the near future, some of the cross-country links are going to have to be on HF. While HP links are not as reliable as VHF links, we simply cannot expect to have a complete chain of VHF sites across the country as soon as we need it. The HF links will have problems not encountered on VHF links, such as long-term fading and multipath effects. To combat these problems, Bob Watson, and Paul Rinaldo, W4RI, designed the Packet Adaptive Modem (PAM).

Briefly, PAM uses a 6ØØ-Hz shift at 75, 15Ø, 3ØØ, 6ØØ, or 12ØØ bauds. Programmable switched-capacitor filters (like those used on the TAPR TNC) keep the modem passband as narrow as possible, reducing noise and QRH on the received signal. In operation, the stations using PAM can determine the highest data rate that the link can support, and then use that data rate. If the link gets vorse, they can go slower. If the link gets better, they can adapt and go faster. This should allow the use of the best transmission rate for a given link. [For further discussion of PAM, see Second ARRL Amateur Radio Computer Networking Conference proceedings, published by the ARRL.]

PAH is built on an S-1ØØ card, but only uses the S-1ØØ pover supply. Serial I/O an:! filter control

Jon Bloom, KE3Z, has built two PAMs in the ARRL lab. One of these "alpha test" modems will be sent to Ralph Wallio, WØRPK, and the other will be set up at W1AW. Testing of the modems will then be conducted on several HF bands.

After these units are tested, the circuit board layout will be updated and several "beta test" modems will be made available for more videspread testing.

RMPRA PACKET BULLETIN BOARD

The Rocky Mountain Packet Radio Association (RMPRA) is now operating what they believe to be the most sophisticated packet bulletin board around.

The system is a modified version of The Bread Board System (TBBS), by Phil Becker, WBØEIV. It was donated to RMPRA and modified for packet-radio operation by Phil and Dave Ebert, W7RH.

TBBS is a modular software system that can be easily modified by the system operator to fit his particular needs. The RMPRA version is accessable through the Pike's Peak digipeater or via telephone at (3Ø3)-452-4735. If you want to call up and see the softvare in operation, set your terminal to 3ØØ bauds, half duplex, 8 data bits and no parity.

23-CM BAND PLANS AND PACKET RADIO

Although most packet-radio activity is now on the tvo-meter band, band plans for the higher VHF and UHF bands will have profound effects on the future of packet radio. The number of packet stations on the air increases daily, and the need for high-speed, videband links is becoming obvious in many urban areas. Both of these factors will push packet radio onto the higher bands, where several hundred kiloHertz of band space should be easy to come by.

Since it is important for packet-radio enthusiasts to keep track of VHF/UHF spectrum plans, we present here a proposed band plan for the 23 cm band.

The ARRL VHF UHF Advisory Committee will recommend a 23 cm bandplan to the ARRL Board of Directors at the B.O.D.'s October meeting. The latest draft of this recommendation contains the following allocations of interest to packet-radio operators:

1249 MHz - 1258 MHz	Wideband communications
1258 MHz - 1260 MHz	Medium bandwidth digital duplex (with 1288- 1290).
1275.5 MHz - 1276 MHz	Single-frequency digital communications.
1276 MHz - 1288 MHz	Wideband communications.
1288 MHz - 1298 MHz	Medium-bandwidth digital duplex (with 1258 - 126Ø).
1297 MHz - 13ØØ MHz •	Single-frequency communications, eg. digital, control links, cross-mode and remote base.

"Wideband communications" includes ATV, spread spectrum, and digital. Those portions of the band allocated to wideband communications will be coordinated by region. It is further recommended that coordination of multiple users of a single channel in local areas can be achieved through isolation by means of cross polarization and directional beam antennas.

It should be remembered that the above is a draft of a recommendation that will be presented to the ARRL Board of Directors. It is not a currently adopted bandplan.

It is important that packet-radio enthusiasts in touch with representatives of their local and nation81 frequency-coordination groups 60 that packet future will be provided for. The ARRL Board of Director'6 meeting will be in October, so be sure to express your feelings on 23-cm bandplan to your Director.

CLUB LISTING CORRECTIONS, ADDITIONS

Add to the listing of packet-radio clubs:

Oahu Packet Enthusiasts Club (OPEC)
P.O. Box 1355
Pearl City, HI 96782

Pacific Packet Radio Society (PPRS)
c/o Hank Magnuski, KA6M
311 Stanford Avenue
Menlo Park, CA 94025

Change these addresses:

Florida Amateur Digital Communications Association (FADCA)
c/o Gwyn Reedy, W1BEL
812 Childers Loop
Brandon, FL 33511

Rocky Mountain Pakcet Radio Association (RMPRA)
c/o Andy Freeborn, NØCCZ
5222 Borrego Drive
Colorado Springs, CO 8Ø918

TAPR GETS OFFICE

Tucson Amateur Packet Radio (TAPR), an international Amateur packet radio research and development group based in Tucson, Arizona, is proud to announce the opening of its office.

The office address is:

TAPR
1Ø16 East Pennsylvania Avenue
Suite 362
Tucson, AZ 85714
Phone: (6Ø2) 746-1166
Office hours are 9:ØØ AM - 5:3Ø PM (MST) Monday through Friday.

The office staff is provided to expedite information requests, provide spare parts support, fill orders, etc. They can NOT ansver technical questions. Technical questions should be routed to the TAPR P.O. box (printed in the last issue of Gateway).

AEA PACKET RADIO INFORMATION

John Gates, N7BTI, of Advanced Electronic Application6 is willing to talk on the phone or correspond by mail to ansver technical questions concerning packet radio.

Contact: AEA
P.O. Box C2160
Bldg O&P 2006-196th SW
Lynnvood, WA 98036-0918

PACKET RADIO AT HAMFESTS

Several Buffalo, NY, ham radio club6 will be sponsoring the HAM-O-RAMA at the Erie County Fairgrounds on September 7 and 8. This hamfest is of intrest to packeteers, since the swap meet will feature computer equipment and the presentations will address "Computers and Amateur Radio." Jeff Ward of 'the ARRL will give an introduction to digital communications at 9:30 Saturday morning, and Gil Bolke, of GLB Electronics, will discuss packet radio later on Saturday.

REPRODUCTION OF GATEWAY MATERIAL

Material may be excerpted from Gateway without prior permission, provided that the original contributor is credited and Gateway is identified a6 the source.

Gateway: The ARRL Packet Radio Newsletter is published by the

American Radio Relay League,
225 Main Street
Nevington, CT 06111

Jeffrey W. Ward, K8KA
Editor

Larry E. Price , W4RA
President

David Sumner, K1ZZ
General Manager

Subscriptions are available to ARRL member6 and nonmember6, at the following rates:

For 25 issues of Gateway:
In the U.S.
ARRL Member $6.00
Nonmember $9.00
In U.S., Canada, and Mexico by First Class Mail
ARRL Member $11 .OO
Nonmember $14.00
Elsewhere by Airmail
ARRL Member $14.00
Nonmember $17.00.

Price6 are subject to change without notice. Remittance must be in U.S. funds, and checks must be drawn on a bank in the U.S.

Gateway: The ARRL
Packet Radio Newsletter
American Radio Relay League
225 Main St.
Nevington, CT 06111

Nonprofit Organization
U.S. Postage
PAID
Hartford, Conn.
Permit No. 2929

Gateway

Vol. 1, No. 3
Sept. 11, 1984

The ARRL Packet-Radio Newsletter

PACKET AT THE BUFFALO HAM-O-RAMA

I just returned from the Buffalo Ham-0-Rama, a large swap meet and convention hosted by several radio clubs in Buffalo, NY and neighboring Ontario. Packet radio was very well represented, both on the technical presentation schedule, and in the indoor display area.

On Saturday morning, I gave a well-attended "Introduction to Digital Communications." I finished the talk with a short discussion of packet radio, and several people in the audience requested more information on packet. Later in the afternoon, Gil Boelke, W2EUP, from GLB electronics, held an in-depth packet-radio seminar and demonstration. It looked like about seventy people attended Gil's talk.

At the GLB booth, a couple of GLB PK1 TNCs vere on-the-air. Both of the TNCs were connected to computer6 running GLB's CP/K program. This is a CP/M program that makes it easy to control the PK1. CP/K provides a menu of commands for the PK1, and also facilitates remote operation of CP/M computer6 attached to PK1 TNCs. If you have a PK1, contact GLB to find out more about the CP/K software .

In the booth next to GLB's was Bob Richardson, W4UCH, from Richcraft Engineering. Richcraft market6 TNC software that runs on Radio Shack computers. (Watch for an upcoming QST article by Bob explaining his "software approach" to packet radio.) A program demonstrating Richcraft software and explaining packet radio was running on a TRS-8Ø at the Richcraft booth.

The number of people at the talks on digital communication6 and at the two packet-radio booths made it clear that more and more amateur radio operator6 are becoming interested in packet radio.

ST. LOUIS PACKET RADIO CLUB

No slight was intended when I left the St. Louis Packet Radio Club (SLAPR) off the Gateway list of packet-radio clubs. The omission brought a letter from Spencer Branham, KAØIXI, reminding me about SLAPR and describing the club's activities. What better opportunity to print the first Gateway packet-radio club profile?

SLAPR was formed in the summer of 1982 by Pete Eaton, WB9FLW. Pete was one of the mainsprings of the Tucson Amateur Packet Radio group during the design and implementation of the TAPE TNC. The PC board layout for the TAPR Beta-test TNC vas done in St. Louis, and the custom power transformers for the TNC were manufactured there, SLAPR was one of the Beta-test sites for the TAPR TNC, and club members have been on-the-air since the first TNCs arrived in January, 1983.

Until 1983, SLAPR publiehed a bi-monthly newsletter with over 1ØØ subscribers. Then, with many stations on-the-air, the printed newsletter gave way to electronic messages posted on packet-radio bulletin boards. There now 3 bulletin boards operated by members of SLAPR.

The network around St. Louis consists of about 3Ø stations, and digipeaters link the group as far north as Peoria, IL. Recently, one St. Louio station was able to connect to a station in Ames, Iowa.

For further information about SLAPR, contact:
SLAPR
c/o Spencer T. Branham
9926 Levis and Clark Blvd.
St. Louis, MO 63136.

NEW FLORIDA PACKET CLUB

The organizing meeting of the Central Florida Packet Association wa6 held on August 22. Jim Diggs, K4AHO; Joseph Leavitt, KF4WM; Larry Gilbreath, WD4BMN; Ed Cox, WØRAO; and Butch Weber, WA4GIF formed the club. There are 1Ø charter memberships available to organizations or individuals. For further information, contact
Jim Diggs, K4AHO
79ØØ Plunkett Ave.
Orlando, FL 3281Ø.

via FADCA>BEACON

ARRL DIGITAL COMMUNICATIONS COMMITTEE

At a meeting in 1981, the ARRL Board of Directora asked then-ARRL President Harry Dannals to forr "an ad hoc committee to recommend standard6 for digital communication6 in the Amateur Radio Service." Preoident Dannals and the next ARRL President, Vic Clark, soon completed the formation of the ARRL Ad Hoc Committee on Digital Communication. The "Digital Committee" advises the ARRL Board of Director6 on matter6 concerning digital communications, and ha6 been involved in such issues as the legalization of AMTOR, and the standardization of packet-radio protocols.

If you have information that you think ahould be brought to the attention of the Digital Committee, contact one of the following Committee members: Paul Rinaldo, W4RI (Chairman); Dennis Connors,

KD2S; Terry Fox, WB4JFI; Douglas Lockhart, VE7APU; Welly Linstruth, WA6JPR; Dr. Henry S. Magauski, KA6M; Paul Newland, AD7I; Eric Scace, K3NA. The members of the Digital Committee are amateurs with experience and expertise in digital communications. When requesting information or guidance from one of them, remember that they are all volunteers, and they do not have time to answer every question and discuss every issue.

DIGITAL COMMITTEE MEETING PREVIEW

During the weekend of September 15 and 16, the Digital Committee will meet at ARRL Headquarters to discuss several topics of interest to Gateway readers.

The first order of business at this meeting will be to inspect and approve the specification of AX.25 Level Two (link level) put forth by Terry Pox, WB4JFI, of AMRAD. Approval of this specification by the Digital Committee is the last step necessary before the protocol is "officially" approved by the ARRL Board of Directors. The protocol described in this document is essentially the same as, and completely compatible with, the protocol that most of us use in every-day packet operation. Once this protocol has been made official, it will be published by the ARRL so that foreign Amateur societies, equipment manufacturers and digital experimenters will be able to work vith a rigorous definition of AX.25.

The approval of AX.25 Level Two will probably be a routine matter, and most of the weekend will be devoted to discussion of network and internetwork protocol6 for Amateur Radio. A network protocol manages communications between stations in the same local network, and is called a "Level 3a" protocol. An internetwork protocol manages communication6 among local networks, and is designated "Level 3b."

The need for Level 3a is obvious to anyone who uses a crowded local network. Without Level 3a, stations that can't hear each other use digipeaters to communicate. The digipeater is a simple device; if it hears a packet that it is to repeat, it repeats the packet. If that packet gets lost in a collision, the sending station must retransmit the packet, and the digipeater must repeat the packet again. When Level 3a is implemented, some digipeaters will be replaced by network controllers. These network controllers will not only forward packets, but make sure that the forwarded packets are delivered. If a packet forwarded from a network controller gets destroyed, the network controller, and not the sending station, will retransmit the packet. This should greatly enhance local communications by reducing the number of collisions on the network.

The need for a Level 3b protocol is equally obvious. We want to send packets from coast to coast and from country to country. It is the internet layer, which forwards packet6 through series of local networks, that will allow us to do this.

Proposals and comments presented at the upcoming Digital Committee meeting will have direct effect on the development of amateur packet radio. The next issue of Gateway will summarize the Digital Committee meeting.

ATLANTA PACKET REPEATER

In late August, a duplex repeater dedicated to packet radio began operation on Black Jack Mountain near Atlanta, GA. The repeater operates on 146.13/.73 MHz.

Via Sandy Donahue, WA4ABY

CITS/ARRL TECHNICAL SEMINAR

The Central Iowa Technical Society (CITS) will host the first annual ARRL Iowa Section technical seminar on September 22 in Des Moines. The seminar runs from 9:ØØ AM to 5:ØØ PM, and will feature packet radio.

Lyle Johnson, WA7GXD, Tucson Amateur Packet Radio (TAPR) president, will discuss development of TAPR terminal node controller (TNC) software and TAPR participation in current linking discussions. Lyle was one of the prime-movers during the design, testing, and production of the TAPR TNC. He received the 1984 Dayton Hamvention Technical Achievement Award for his work on the TNC project.

Jeff Ward, K8KA, from the ARRL will give an overview of ARRL technical activities, with emphasis on recent digital communications projects.

John Maurer, NAØS, of CITS, will present last-minute details on the CITS computer bulletin board system (CBBS) hardware and software. John, and Dave Huffman, KAØDNB, will also discuss progress toward an inexpensive packet-radio link management unit based on surplus microcomputer hardware.

Finally, Ralph Wallio, WØRPK, will review the background, success, and future plans associated with packet-radio meteor-scatter communications.

Proceedings will be made available after the conference. For more information, contact Ralph Wallio, WØRPK, CITS President, at (515)961-64Ø6. via WØRPK.

METEOR BURST SUCCESS

Ralph Wallio, WØRPK, of Indianola, Iowa, and Robert Carpenter, W3OTC, of Rockville, Maryland, have been making routine packet-radio contacts via 6-meter meteor-burst propagation. WØRPK uses 25Ø W into a S-element beam, and W3OTC uses 15Ø W and a 6-element beam. Both stations are using AFSK FM at 12ØØ bauds. In order to take advantage of short meteor trails, packets are limited to 3Ø or 4Ø characters of information. Taking overhead (addressing and other protocol information) into consideration, the information packets are between 32Ø and 38Ø ms long. Acknowledgement packets are 12Ø ms long. This should allow transfer of information on meteor trails with duration between 44Ø and 5ØØ ms. Unfortunately, WØRPK's transceiver need6 1ØØ ms for T/R switching, bringing the necessary trail duration to about 6ØØ ms. Even though this system is not optimized, several long file6 were transferred. Throughput was about 1 bit/s.

Work is now under way to achieve better throughput. Possible improvements include more efficient modulation techniques, higher output power and special protocols. Via W3OTC.

SOUTHERN CALIFORNIA 23-CM BAND PLAN

On August 25, the Southern California Repeater and Remote Base Association (SCRRBA) hosted a public meeting to discuss a new southern California band plan for the 23-cm band. Each group of 23-cm users, including packet-radio experimenters, chose a spokesman. After a +-hour discussion, a band plan was agreed upon.

Those sections of the band all0cated to digital communications are listed below.

1246-1248	Narrow-band FM and narrow-band digital (<5Ø-kHz bandwidth).
1248-1251.5	Wideband digital (>5ØØ-kHz bandwidth).
1258-1260	Narrow-band FM and narrow-band digital (<5Ø-kHz bandwidth).
1260-1270	Satellite uplink and non-coordinated simplex (including experimental wideband techniques).
1288-1294	Non-coordinated simplex (including experimental wideband techniques).
1297-1300	Wideband digital (>5ØØ-kHz bandwidth).

This band plan was prepared specifically for use in the southern California geographical region. It was unanimously approved by the representatives of ALL the users groups. This band plan takes effect immediately, and will be reviewed after three years.

Since the southern California region leads other areas of the country in 23-cm band activity, SCRRBA believe6 that the new band plan may be of use to other coordination councils. It is important that digital users represent themselves at band-planning meetings, so that the needs of wideband digital communications will be met over the next few years. Via SCRRBA news release.

PACKET-RADIO WEATHER NETWORK?

As packet-radio networks grow, and operators begin to tire of real-time, RTTY-like contacts, the search will start for applications for this new communications tool. A letter recently received by Paul Rinaldo, W4RI, from Dr. Fred Decker, U. S. Deputy Assistant Secretary for Education, points out that a packet-radio network would be a great means of distributing weather information collected at amateur meteorological stations.

Many schools around the country have student-run weather stations and meteorology clubs. Some of these schools also have amateur radio clubs and stations. What better way to introduce interested and capable students to amateur radio (and packet radio) than by using amateur packet networks to connect amateur weather stations? Such a project would blend the natural sciences, computers and amateur radio into an interesting and useful whole. It might bring some young people in contact with amateur radio, and it certainly would provide "users" for our growing packet network.

If you are interested in meteorology and packet radio, or if you are just interested in packet radio and public service, contact Gateway for further information about this interesting possibility.

EMERGENCY COMMUNICATIONS

On July 31, members of the Florida Amateur Digital Communications Association (FADCA) conducted a demonstration of packet-radio emergency communications. At the request of the Martin County Public Safety Department, stations were placed at the Mart in County Emergency Operations Center (EOC), the Dade County EOC, St. Lucie, Palm Beach, and the National Hurricane Center in Miami. Traffic was successfully passed among all stations. As a result of the demonstration, the Martin County EOC will be installing a TNC and a two-meter rig.

If you are planning emergency operation, the FADCA team advises that a standard message format is needed, and each station should be able to print and buffer messages.
Via FADCA>BEACON

TAPR SOFTWARE UPDATES

In the first issue of Gateway, Harold Price, NK6K, a member of TAPR's software development team, discussed some of the projects under way at TAPR. One of the projects mentioned was TAPR TNC software Version 4.0. When asked about the status of this software, Harold made the following comments:

"The scope of TAPR software release version 4.0 may change as a result of the Digital Committee meeting in mid-September. Version 4.0 is intended to allow groups to experiment with network layer protocols, and may require some tweaking based on what class ot network layer protocol receives the most support during the meeting. We remain convinced that no matter what approach is taken, a TNC with the ability to maintain multiple connections at level two will be a big advantage to network developers.

"There are a number of fixes and extensions that are desirable outside the context of Version 4.Ø. A report and request for comments was released to the electronic networks in late August, and will appear in the next edition of the TAPR Packet Status Register (PSR). If Version 4.0 is pushed back, these fixes and extensions will be released as Version 3.4. No date has been set for these releases [So don't bother them about it. Ed.], but they are high priority items." From NK6K

UOSAT DATA COMMUNICATIONS EXPERIMENT

Harold Price, NK6K, also shed some light on the status of the UOSAT-OSCAR 11 Data Communications Experiment (DCE). This is an experimental store-and-forward digital communications system on board uo-11. The primary objectives of the DCE are to test satellite and ground station hardware and software for the upcoming PACSAT mission.

Harold says: "The Digital Communications Experiment on board UOSAT-OSCAR 11 is still in the spacecraft command loop. Once the ground station in Los Angeles is fully commissioned, procedures will be worked out that allow testing of the DCE while still maintaining a high reliability path to the satellite's command system. The DCE is currently being used to bypass a failed component in one of the command uplinks." From NK6K

PACKET RADIO AT THE YORK HAMFEST

On September 22, Jon Bloom, **KE3Z,** an ARRL Laboratory Engineer, will give a taik on packet radio at the York, PA Hamfest. The hamfest is sponsored by several York-area amateur-radio clubs, and take6 place September 22 and 23 at the York Fairgrounds.

SPECIAL SERVICE CLUBS

If you are receiving Gateway because you are an official of an SSC, please make full use of your complimentary subscription. Pass Gateway along to member6 of your club who might be interested in digital communications. Use any interesting Gateway-items in your club newsletter. If your club takes particular interest in packet radio, contact one of the packet-radio clubs listed in the first issue of Gateway. Members of those club6 are usually willing to give demonstrations for interested groups.

REPRODUCTION OF GATEWAY MATERIAL

Material may be excerpted from **Gateway** without prior permission, provided that the original contributor is credited and Gateway is identified as the source.

Gateway: The ARRL Packet Radio Newsletter is published **by the**

American Radio Relay League
225 Main Street
Newington, CT 06111

Jeffrey W. Ward, **K8KA**
Editor

Larry **E.** Price, **W4RA**
President

David Sumner, **K1ZZ**
General Manager

Subscriptions are avmailable to ARRL members and nonmembers, at the following rates:

For **25** issues of Gateway:
- In the U.S.
  - ARRL Member $6 .OO
  - Nonmember $9 **.00**
- In U.S., Canada, and Mexico by First Class Mail
  - ARRL Member $11 .OO
  - Nonmember $14.00
- Elsewhere by Airmail
  - ARRL Member $14.00
  - Nonmember $17.00.

Prices are subject to change without notice. Remittance must be in U.S. **funds,** and checks must be drawn on a bank in the U.S.

Gateway: The **ARRL**
Packet Radio Newsletter
American Radio Relay League
225 Main **St.**
Newington, CT **06111**

Nonprof it Organization
U.S. Postage
PAID
Hartford, Conn.
Permit No. 2929

Gateway

Vol. 1, No. 4
Sept. 25, 1984

The ARRL Packet-Radio Newsletter

ARRL DIGITAL COMMITTEE MEETING

The ARRL Ad Hoc Committee on Amateur Radio Digital Communication met in Newington, Connecticut on September 14, 15 and 16. [See issue 3 of Gateway for background on the Digital Committee.] Twenty packet-radio enthusiasts (committee members and observers) were present for the meeting. The group addressed the state of amateur packet radio; link-layer protocols; and approaches to packet-radio networking.

The following items summarize the results of the weekend's meetings.

DIGITAL COMMITTEE APPROVES AX.25 SPECIFICATION

During its meeting on September 15, the ARRL Ad Hoc Committee on Amateur Radio Digital Communication approved a specification of the AX.25 link-level protocol. This clears the way for submission of the protocol to the ARRL Board of Directors for their approval.

The approved specification of AX.25 was written by Terry Fox, WB4JFI. The protocol is essentially the same as that implemented on all of the TNCs now on the air. Major changes from previous versions of AX.25 concern the poll/final (P/F) bit and multiple digipeaters. The specification includes the P/F bit resolution proposed by Phil Karn, KA9Q [See July, 1984 QEX]. The use of multiple digipeaters, as permitted on most available TNCs, is now part of the "official" protocol. Neither of these changes will make the specified protocol incompatible with existing implementations.

The ARRL Board of Directors will consider AX.25 at its next meeting, in October of this year. Board of Directors' approval will create a published, international standard protocol for amateur packet radio. With such a standard in existence, commercial manufacturers will be able to design products for packet radio, confident that their products will be compatible with others'.

23-CM BAND PLAN COMMENTS

The Digital Committee also considered the matter of UHF band plans. After some discussion, the committee endorsed the ARRL VHF/UHF Advisory Committee (WAC) band plans for both 23 cm and 33 cm. The committee noted that the Southern California Repeater and Remote Base Association (SCRRBA) 23-cm band plan also includes sufficient frequency allocations for digital communication. The digital allocation in the SCRRBA band plan is not on the same frequency as that in the ARRL band plan, but this should only cause slight problems on links going into and out of Southern California.

PACKET-RADIO NETWORKING

The next step in the development of amateur packet radio is the specification and implementation of a network-layer protocol. The network layer is responsible for routing packets from their originator to their addressee through a number of intermediate stations. There are currently two network protocols being proposed for amateur packet radio. To help decide which of these protocols is better for amateurs, the Digital Committee had a "brainstorming" session. The session provided an interesting look at the future of amateur packet radio. Some of the points made during the discussion follow:

- A user should need only a TNC to access the network.
- The network should be easy to operate.
- The network should support a wide variety of media (HF, VHF, microwaves, meteor scatter, etc.).
- The network should remain amateur in spirit.
- The network should work when commercial networks are down; in the event of emergency, the network should still be available.
- Network users should be able to specify their needs (for example, a choice of speed over reliability).
- The network should be able to resist jammers.
- The network should be able to take advantage of new stations when they come on the air, and should not be crippled by stations leaving the air,
- The network should support traditional amateur operating modes, rather than forcing amateurs to change their habits.
- Both real-time and store-and-forward operation should be supported.
- Controlled broadcast, or alert should be available.
- Roundtable operation should be supported.

- There should be a network directory available on the air.
- Fault detection and isolation should be easy.
- The network should function well in various geographical regions.
- There should be a mechanism for monitoring the performance of the network.
- There should be no user "passwords" or other "authentication."
- The network should be data transparent; whatever data is sent from the source should be received unchanged by the destination.
- The network should be global.
- The goals set for the network should be attainable within the limited, amateur resources available.

This is an ambitious set of goals. The Digital Committee feels that, given the proper approach and efficient use of available **resources,** these goals are attainable.

After the brainstorming session, the committee heard tutorials on the two proposed network approaches, data&rams and virtual circuits. A very brief summary of each approach follows.

Each packet sent through a **datagram** network is routed independently of any preceding or following packets. This routing system is very robust; if stations join the network, they can be immediately taken advantage of. If stations leave the network (as long as a path exists from the source station to the destination station) communications will not be interrupted. There is a price which must be paid for this robustness: each packet must contain extensive information needed for routing. The **datagram** protocol being advanced for amateur packet radio is the Defense Advanced Research Projects Agency **(DARPA)** Internet Protocol **(IP).**

In a virtual-circuit network, a route is established between the calling station and the called station at the beginning of a QSO. This route is called a virtual circuit or a virtual connection, and is used by all of the packets in the QSO. Only the first or call-setup packet must contain complete routing information. Once the virtual circuit is established, subsequent packets need only include a virtual-circuit number. If one of the stations in the virtual **circuit becomes** unable to handle traffic, another call-setup packet must be sent, and another virtual circuit set up. Note that most packets will not be **call-**setup packets, and will contain minimal routing **informat** ion. This is one of the advantages of the virtual-circuit approach to networking. The virtual circuit protocol proposed for amateur use is called AX.25, and is set forth by Terry Fox, **WB4JFI,** in the Third ARRL Amateur Radio Computer Networking Conference proceedings.

These descriptions do not fully describe the protocols under consideration, nor do theycapture the scope of the discussions that took place during the Digital Committee meeting. After almost two days of discussion, there were still proponents of virtual circuits, proponents **of** datagrams, and a number of undecided individuals.

The Digital Committee has no desire to force a protocol upon any group. Because of the number of people in the undecided camp and the number of points in favor of each type of protocol the committee decided that the packet community would best be served by a period of experimentation, during which both datagrams and virtual circuits would be implemented and tested.

Several members of the committee stressed the importance of implementing the protocols in a transportable high-level computer language. This would allow the programs that are developed to be run on many computers. The people who are going to be doing the programming decided to use the **"C"** programming language. In order to test the two protocol implementations, they must operate in the same environment. The test environment chosen by the committee is the Xerox 820 computer, modified to use an HDLC chip and run at **4** MHz. These decisions are not meant to stifle network protocol experiments. They were intended to foster program transportability, and to provide a common environment for testing protocol performance.

The identification of network objectives, the discussion of protocols, and the formulation of an experimental approach are the first steps toward the implementation of a world-wide packet-radio network. If you are interested in the network protocol experiments, contact a member of the ARRL Digital Committee [See Gateway f3.1 and get started!

EUROPEAN PACKET ACTIVITY

Several European observers were present for the ARRL Digital committee meeting. They were: Hanspeter **Kuhlen, DK1YQ,** from **AMSAT-DL;** Ian Wade, **G3NRW,** from the British Amateur Radio Teleprinter Group (BARTG); John Sager, **G8ONH** from the RSGB; and Alan Jones, **G8WJM,** from the Cambridge Computer Labs. The primary duty of these observers was to learn about the state of amateur packet radio in the U.S., but they also provided us with some news about packet radio overseas.

Germany: In West Germany, several TAPR **TNCs** have made it through customs into the hands of the new Bavarian Packet Radio Group. There should be twenty or more packet stations on in West Germany by the time you read this. Unfortunately, there are some German regulations that will slow the development of packet networks; no two repeaters may be linked together, and no third-party traffic may be passed. Hanspeter was confident that these hurdles could be overcome. For more **informat** ion on the Bavarian Packet Radio **Group,** contact:

Thomas Hieselbach, **DL2MDE**
Narzissenweg **1Ø**
D-8031 WESSLTNG
W-Germany.

England: The RSGB, in conjunction with BARTG, has formed a Packet Radio Working Party. The aims of the group, chaired by Dr. Dain Evans, **G3RPE,** are **:** to clarify what is meant by Packet Radio in the amateur context; to determine what form Packet Radio is likely to take in the UK, and to compare this with activity in other countries; to determine how packet repeater proposals are to be handled; and to determine the requirements for

possible changes of the license regulations to exploit the capabilities of Packet Radio to the full. The Working Party has held two meetings, and adopted AX.25 as a standard for the UK. The BARTG newsletter Datacom is a good *source* of information on packet radio in the UK. Write:

Ian Wade, G3NRW
7 Daubeney Close
Har ling ton
Dunstable
Bedfordshire
LU5 6NF
UK.

There are already thirteen stations running AX.25 packet radio in the UK, and several dozen others are actively interested. Expect to hear UK packet stations on Oscar-10 or HF any day now.

Via DATACOM

PACKET RADIO ON JAPANESE SATELLITE

Sometime between 1985 and 1987, the Japan Amateur Radio League (JARL) and JAMSAT will be launching the first all-Japanese amateur satellite, JAS-1. Along with a Mode-J transponder, JAS-l will carry a store-and-forward packet-radio transponder. The packet-radio transponder will operate at 12ØØ bauds, FSK, with an uplink in the Z-meter band, and a downlink in the 7Ø-cm band. The designers of the satellite have agreed to use the AX.25 link-layer protocol, strengthening the protocol's standing as an international standard.

PHASE III C AND PACKET RADIO

If the amateur satellite fraternity is able to fund the launch of a Phase-III C satellite, that satellite might carry a packet radio transponder. The ability of the satellite to carry such a transponder will be regulated by three factors: kick motor capacity, space available, and power available. If the proposed steam-propelled kick motor is used, there should be both the motor capacity and space available to launch a digital transponder. The power-consumption limits placed on the design of the transponder would make most of the existing technology useless, since existing high-level data link control (HDLC) chips draw too much current for satellite application. The use of software to transmit and receive packets through CMOS serial I/O devices is under considerat ion. A packet transponder for Phase-III C is not a high-priority item for AMSAT-DL, but it is under consideration. Via DKlYQ

PACKET COLUMN IN RADIOSPORTIRG

Puri Blanarovich, VE3BMV is hoping to start a new amateur-radio magazine called Radiosporting. He plans to make it "for active radioamateurs, filled with timely information that would be of interest to many involved in operating and competitive aspects of amateur radio." One of the columns in **Radiosporting** will be devoted to RTTY, AMTOR and packet radio. If you are interested in subscribing to Radiosport, or contributing to its packet column, contact:

RADIOSPORTXNG
Box 65
Don Mills, ON, M3C 2R6
CANADA .

MORE NEWS ON HIGH-SPEED PACKETS

We received the following letter from Richard Bisbey II, NG6Q:

"The Information Sciences Institute Amateur Radio Group (WB6MXZ) has been on the air with a 56 Kb packet radio system since December, 1983. The equipment consists of 8Ø88-based controllers using the Zilog 8530 SCC chip and frequency agile 1Ø-watt FSK transmitters and receivers. The controller supports serial, parallel, and Ethernet connected devices. Assembled controllers cost approximately $650. The system uses standard INTERNET (IP/TCP) protocol, and in addition to supporting simple link level connections, supports internetworking as well as numerous high-level protocols such as TELNET, File Transfer (FTP), Mail (SMTP), Multi-Media Mail (MMM,MPM), Graphics (GP), Packet Voice (NVP), and Remote Virtual Disk (RVD). The hardware and software can support data rates as high as 1 Mb, well in excess of the current amateur limit of 56 Kb.

"Previous packet radio accomplishments by ISI amateurs include a March 1982 demonstration of transcontinental packet radio internetworking between an aircraft flying above Los Angeles and a fixed station in Virginia, and a November, 1982, demonstration of intercontinental internetwork packet radio communications between Los Angeles, California and The Hague, Netherlands using Intelsat-4A. The ISI group is currently developing a low-cost spread-spectrum system with capabilities similar to the current non-spread system."

SEATTLE, WA PACKET CLUB

The Seattle-Tacoma area packet radio users recently decided to become a formal organization. The actual formation of the club will take place October 13, 1984, at 13ØØ Pacific Time in the Boeing Employees' Amateur Radio Society meeting house. The club constitution will be a slightly modified version of the NEPRA constitution. The first large-scale goal of the club will be linking of the Seattle-Tacoma-Everett area to Vancouver, BC; Portland, OR; Spokane, WA; and Moscow, ID. Via KD7UW

PACKET RADIO IN GEORGIA

The Georgia Radio Amateur Packet Enthusiasts (GRAPES) club was formed on September 14. Their network name? GRAPEVINE, of course. They are interested in setting up repeaters to link Georgia with the Carolinas, Florida, Alabama and Tennessee. They would like to hear from packeteers in those areas. One possibility that they are examining is a repeater location on Brasstown Bald, a 4,784-ft mountain on the northern Georgia border. For informat ion, contact:

Dennis Barrow, WB4GQX
Rte 7 Heard Road
Cumming , GA 30130. Via W4RI

ABSTRACTS

As space permits, and interesting articles come to our attention, Gateway will print abstracts of articles found in other journals and newsletters.

If you know of an article that should be abstracted, drop us a note and a copy of the article.

"HF Packet Frequency Specification," by John Langner, WB2OSZ; NEPRA Packetear, Issue 14. This one-pager addresses the problem of frequency specification for FSH stations on HF. It proposes that the radio frequency midway between the mark and space frequencies be used when discussing the operating frequency of such stations.

"Routing Strategies in Computer Networks," by Hsieh and Gitman; Computer, June, 1984. This article addresses the question of routing. It sets forth some algorithms for determining the best route from one node to another, and discusses the application of these algorithms to various types of computer networks. Since it addresses routing in both datagram and virtual-circuit networks, this article is particularly interesting to people concerned with amateur packet-radio networking.

REPRODUCTION OF **GATEWAY MATERIAL**

Gateway: The ARRL Packet Radio Newsletter is published by the

American Radio Relay League
225 Main Street
Newington, CT 06111

Jeffrey W. Ward, K8KA
Editor

Larry E. Price, W4RA
President

David Sumner, K1ZZ
General Manager

Subscriptions are available to ARRL members and nonmembers, at the following rates:

For 25 issues of Gateway:
- In the U.S.
  - ARRL Member $6 .OO
  - Nonmember $9 .OO
- In U.S., Canada, and Mexico by First Class Mail
  - ARRL Member $11.00
  - Nonmember $14.00
- Elsewhere by Airmail
  - ARRL Member $14.00
  - Nonmember $17.00.

Prices are subject to change without notice. Remittance must be in U.S. funds, and checks must be drawn on a bank in the U.S.

Gateway: The ARRL
Packet Radio Newsletter
American Radio Relay League
225 Main St,
Newington, CT 06111

Nonprofit Organization
U.S. Postage
PAID
Hartford, Conn.
Permit No. 2929

Gateway

Vol. 1, No. 5
Oct. 9, 1984

The ARRL Packet-Radio Newsletter

IOWA SECTION TECHNICAL SEMINAR

There was not enough room in Gateway #4 for a report on my trip to Iowa, September 21 and 22, for the Central Iowa Technical Society's Iowa Section Technical Seminar. The seminar covered many technical topics, but concentrated on packet radio.

Lyle Johnson, WA7GXD, president of Tucson Amateur Packet Radio (TAPR), gave a well-received presentation on packet-radio linking and the outcome of the ARRL Digital Committee meeting. One of Lyle's points bears repitition: We have begun to experience an unfortunate problem in packet radio. People who would otherwise be experimenting with protocols, or modems, or TNCs are not, because they have heard that "TAPR is going to do it." Lyle is worried that the success of the TAPR TNC project is keeping others from experimenting in packet radio. It must be stressed that packet radio needs experimentation. The question of what networking protocol is best for amateur packet radio is not going to be answered by the Digital Committee, TAPR or any other single organization. The question is going to be answered by experimentation. If you want to experiment, do it. Who knows? It might be your software or hardware that answers a critical packet-networking question.

While I was in Iowa, I got to look at the CITS club station that is taking part in packet meteor-scatter experiments. It is interesting to note that there was no expensive equipment used in the station. A crew from CITS salvaged an old, surplus Motorola VHF rig, built and erected a six-meter beam, and interfaced it all to a TAPR TNC. The only parts that had to be purchased for the project were six-meter crystals. That is ham radio.

VHF AND UHF EXPERIMENTATION

Giving a talk at the Mid-Atlantic States VHF Conference, I was reminded that packet radio, while it makes heavy use of digital technology, has a lot to offer amateurs who are not interested in computers. If you are a VHF or UHF experimenter, you will find that packet radio is fertile ground for VHF and UHF applications. In many parts of the country, the two-meter band is full, and the 220-MHz band is not far behind. This crowding, and transmission speed restrictions on the lower VHF bands, will force packet radio into the UHF and microwave spectrum. So, if you are interested in VHF and UHF design, maybe your local packet group can give you some project ideas.

EASTNET KEEPS GROWING

The east-coast packet radio link has taken one more step toward reality. The MD/DC/VA to Philadelphia link is improving by the day. Joe Fisher, KC2TN, should get a new antenna up this weekend which will make the path as solid as a rock. We have been linking the packet bulletin board systems (PBBSs) at W3IWI and WB2MNF almost nightly. Until we get Level 3 protocols up and running, it looks like linked PBBSs will provide a store-and-forward capability not unlike what we will have with PACSAT. Both Tom Clark, W3IWI and Jon Pearce, WB2MNF, are running PBBSs with Xerox 820 computers and WØRLI PBBS software. From W3IWI

NETWORKING IN THE MIDWEST

In late September, Chicago, IL, and St. Louis, MO were connected by VHF packet radio. The path is in excess of 300 miles, and there are no mountain tops to take advantage of. Congratulations to the more than half-dozen stations involved in the digipeating. Via WB9FLW

UO-11 DATA COMMUNICATIONS EXPERIMENT

During the week of October 1, the UoSat command station in Surrey, England conducted tests and experiments on the data communications experiment (DCE) aboard UO-11. Tests of the VHF uplink and UHF downlink for the DCE culminated with successful reception of 9600-bit/s data on Friday, October 5. Congratulations to the team at Surrey and the rest of the crew responsible for the DCE! We also would like to wish UoSat-OSCAR 9 a happy third birthday. UO-9 was launched on October 6, 1981, and is still in service.

PACKET RADIO FOR EMERGENCY COORDINATORS

The new ARRL Emergency Coordinator's Handbook, with a 20-page chapter on packet radio, is available for $5.00 from the ARRL. This book is "intended to help you acquire, develop and refine the skills which you need to function effectively in serving the public through Amateur Radio communications." The chapter on packet radio should make those in the Amateur Radio Emergency Service (ARES) aware of the traffic handling and organizing capability of packet radio. If your packet club is looking for somewhere to put on a demonstration, try an ARES meeting. If you are an ARES organization looking for an interesting meeting program, contact a nearby packet club.

MORE ON A POSSIBLE PACKET WEATHER NET

Fred W. Decker, W7ANX, U.S. Deputy Assistant Secretary for Education, has said previously that he believes packet radio and amateur meteorology can be united to form a packet-radio weather network [See Gateway #3.]. Such a network would provide some users for the growing amateur packet network, and perhaps generate some interest in amateur radio among amateur meteorologists. In a paper delivered at the First International Conference on School and Popular Meterological Education, Dr. Decker goes on to say:

"(1) Packet radio communication has emerged for amateur use as new manufacturers offer equipment enab 1 ing amateurs to communicate using their computers and [VHF radio equipment]. (2) Repeaters have recently carried packet signals 400 miles in 5 repeater jumps. This opens the possibility of local networks exchanging data at substantial distances. (3) Weather sensors feeding computers and thereby archiving data for interrogation by packet radio have come on the market from at least two firms seeing the potential in computerized popular meteorology (Heath Co. and Vaisala). (4) The drive to "Save 22Ø" among radio amateurs seeks to populate the allocated 22Ø-MHz amateur radio band and thereby fend off commercial intrusion, and the belief grows that weather amateurs and schools can help this cause by adopting this band (1 1/4 meters) for packet radio exchange of computerized school and amateur weather station data. With basic technical innovation completed, it remains necessary only to adopt convent ions...to maximize the ease of data intergchange..."

This is an interesting project, and the fact that Dr. Decker has been discussing it with his col leagues has undoubtedly brought packet radio to the attention of many people in academia. Dr. Noel Petit, WBØVGI, of Augsburg College, has proposed to build a computerized weather network that would communicate via telephone. The cost of his weather stations, including computer, sensors and modem, is about $300. Perhaps, with the help of some packet-radio enthusiasts, Dr. Petit's weather network can use packet radio.

If you are interested in this project, or in giving an educational demonstration of packet radio' contact:

Dr. Noel Petit
Suite 228-B
511 11th Ave. S.
Minneapolis, MN 55415,

or

Dr. Fred W. Decker
Deputy Assistand Secretary for Education
Suite 722, Brown Bldg.
12ØØ 19th Streed, NW
Washington, DC, 2Ø2Ø8.

SOFTNET NEWS

We recently received an issue of Softnet News, the newsletter of the Softnet Users Group (S.U.G.). S.U.G. is a group experimenting with packet radio in Linkoping, Sweden. "The main concept behind SOFTNET is that all packets are considered to be programs in a network language. These programs are interpreted in the nodes as soon as they arrive... This approach makes it possible for a user to define his own high level services like datagrams, virtual calls, file transfers and mailboxes." [From "SOFTNET - An Approach to High Level Packet Communication", Second ARRL Computer Networking Conference.]

The SOFTNET approach could be used, with the AK.25 link protocol, to provide network services in the growing North American packet network.

SOFTNET nodes are currently implemented on four PC boards, a SOFTNODE computer board, a SOFTMEM memory board, a SOFTLINK link and modem board, and a radio board. S.U.G. is now se 1 ling PC boards, PROMS, and software for SOFTNET nodes. The builder must supply the other components needed to populate the boards. Total cost is about $600 per node. Of particular interest to those of us who operate on 12ØØ bit/s local networks is the fact that SOFTLINK radios/modems run at 1ØØ Kbit/s.

The SOFTNET concept is primarily concerned with network and internetwork functions. The SOFTNET News contains two articles on network routing, searching and addressing. S.U.G. is a growing group, moving forward quickly with packet-radio network experimentation. Their progress is sure to help packet experimenters worldwide.

For information, contact: SOFTNET Users Group
Dept. of E.E.
Linkoping University,
S-581 83 Linkoping
SWEDEN

Via SOFTNET News.

TELEPORT STA INFORMATION

The waiver for automatic teleport operation requested jointly by the ARRL and AMSAT should be issued about October 17. This special temporary authority (STA) allows the operation of unattended digital gateway stations for store-and-forward or real-time satellite links. For a complete description of the STA, see Gateway issue 2. Two more stations, N2EKH and Theodore Harris, N6IIU, have been added to the list of participants, bringing the total number of authorized stations to 21. From W1UED

TECHNICAL COORDINATORS AND PACKET

The ARRL Technical Coordinators (TCs) answer technical questions, put on technical talks, and perform many technical functions within their ARRL section. Several notes in the fall issue of the Technical Coordinator news letter indicate that many TCs are interested in packet radio. Richard Regent, K9GDF, editor of the newsletter, has received the following comments from other TCs: "[Our] greatest successful TC program is talking about data communications," and "[we have been] getting packet radio started by helping hams assemble and operate kits, and holding Z-meter packet nets."

If you are interested in finding out more about packet radio, contact the ARRL to get the name of your section's TC. Via Technical Coordinator.

NEBRASKA PACKET GROUP?

Lyman Nelson, WBØIEN, is interested in forming a state-wide packet radio group in Nebraska. Contact:

Lyman Nelson , WBØIEN
Rt. 2
Hooper, NE, 68031.

Via Technical Coordinator.

MODEM FILTER FROM EXAR

"Preliminary" data from the EXAR IC company about the XR-21Ø3 FSK modem filter has just arrived at the ARRL. The device is designed to "perform the complete filtering function necessary for a Bell 1Ø3 compatible modem." The chip uses switched-capacitor technology to implement two 6-pole band-pass filters. One of the filters is centered at 117Ø Hz, and one centered at 2125 Hz. This chip might prove useful to those using the Bel 1 1Ø3 protocol for packet radio on HF. Perhaps, by changing the time-base crysta 1, the passbands could be moved to other frequencies. If you experiment with this chip, send us a letter describing your results. Via W4RI.

PACKET FREQUENCY COORDINATION

We have received word from AMRAD president, Terry Fox, WB4JFI, that The Middle Atlantic Amateur FM and Repeater Council (TMARC), on September 24, designated the frequencies 145.Ø1, 145.03 and 145.Ø5 MHz for primary coordination as packet-radio channe ls in their jurisdiction. TMARC covers Maryland, De laware and Northern Virginia. Via W4RI

PACKET DISCUSSION AT THE TROPICAL HAMBOREE

If you are going to be in Florida this spring (planning ahead?), be sure to attend the 1985 Southeastern Division Convention/25th Annual Tropical Hamboree in Miami, Florida on February 2 and 3. An all-day Packet Radio Seminar is tentatively planned for Sunday, the 3rd. Pau 1 Rinaldo, W4RI, Manager of the ARRL Technical Department, and Lyle Johnson, WA7GXD, president of Tucson Amateur Packet Radio, are scheduled to attend the seminar. More information, and a firm schedule for the seminar will be available in December. Via Dade Radio Club, Inc.

DATA BASE OF PACKET USERS

Dear Packet Radio Enthusiasts;

Several local Amateurs and I, representing Central Illinois Packet Radio User Society (CIPRUS) have started a to compile a directory of all stations who are also active on this fantastic new mode. After great acceptance and encouragement from the midwest as well as other areas, what initially was to be only a regional effort has been expanded to include the entire country.

After giving a great deal of thought as to what the best method would be to compile such a data base, we decided to contact you, the packet radio clubs and other leaders in the area of packet radio. Most of what we have compiled to date has come through direct contact with individual stations. We learned that this method involved a lot of unnecessary paper work and correspondence that could be eliminated by contacting large organizations and groups.

In order to contact as many stations as possible and to keep the costs down as much as practical we would like to propose the fol lowing: If you would send us a listing with all of the required information concerning your member stations along with a large manila s.a.s.e (several stamps please), we will, in turn, send out a compiled listing of active packet stations that you can photocopy and distribute to your membership. If you have your membership list in an IBM PC or compatible data base you can submit a diskette with the required information and have the compiled list returned on diskette. The data base is being maintained on an IBM XT in a dBase II file that can usc data from nearly any standard ASCII or WordStar text file (comma delimited prefered).

Since many of us are not able to offer our services to the technical aspect of packet deve lopment, we view this project as one that will al low us to make a small contribution to the field of packet radio. We do not wish to become involved in the publishing business but rather to act only as a central point to compile and distribute data from you. We have been in contact with TAPR and have discussed the possiblity of enclosing a preprinted post card with every TAPR TNC shipped. This would al low the individual station to merely fill out the necessary blanks and return the card to us for inclusion in the directory. If you are aware of any other manufacturer of who would like to help this project, please have them contact us.

In closing I would like to present the f0 11owing points for your consideration and comment:

1. This project is still in its infancy, with only about 250 stations listed to date; therefore we are open to suggestions as to what you feel should be included in such a directory. Speak up now while it is still relatively easy to modify the data base!

2. We are not aware of any other group or individual who has decided to attempt a similar venture. If you know of such a group, please put them in contact with us so that we do not duplicate efforts. We don't care who does it, just so long as it gets done by someone!

3. The directory is currently being printed on a letter quality printer and duplicated on a photocopier. As the directory continues to grow so do the costs involved -- we are already making over 50 copies per month, with over 2Ø pages each. Professional printing is just around the corner. Therefore, since we are a nonprofit amateur radio club, we will not turn down any donations that you would like to make toward paper costs.

4. The format that we are using is flexible but is being currently printed as fol lows:

Listing #1 -- Alphabetic, by callsign:

Callsign; Name; Address; Misc. Info(1)
City, State, Zip; Misc. Info(2).

Listing #2 -- Numeric, by zip code:

Callsign; Name; City; State; Zip Code

The "Misc. Info" columns may contain any items (approx. 15 characters) of special information that you may wish to include. Some ideas that have been used: frequency, OSCAR, mailbox, digipeater, 24-hour operation, grid square, and gateway operation.

Thank You

Gregory L. Smith - N9AGC
c/o CIPRUS
P.O. Box 4143
Peoria, Illinois 61687

[The above letter has been reproduced in full. Send any comments to Mr. Smith, and Gateway.]

REPRODUCTION OF GATEWAY MATERIAL

Material may be excerpted from Gateway without prior permission, provided that the original contributor is credited and Gateway is identified as the source.

Gateway: The ARRL Packet Radio Newsletter is published by the

American Radio Relay League
225 Main Street
Newington, CT 06111

Jeffrey W. Ward, K8KA
Editor

Larry E. Price, W4RA
President

David Sumner, K1ZZ
General Manager

Subscriptions are available to ARRL members and nonmembers, at the following rates:

For 25 issues of Gateway:
- In the U.S.
  - ARRL Member $6.00
  - Nonmember $9.00
- In U.S., Canada, and Mexico by First Class Mail
  - ARRL Member $11.00
  - Nonmember $14.00
- Elsewhere by Airmail
  - ARRL Member $14.00
  - Nonmember $17.00.

Prices are subject to change without notice. Remittance must be in U.S. funds, and checks must be drawn on a bank in the U.S.

Gateway: The ARRL
Packet Radio Newsletter
American Radio Relay League
225 Main St.
Newington, CT 06111

Nonprofit Organization
U.S. Postage
PAID
Hartford, Conn.
Permit No. 2929

Gateway

Vol. 1, No. 6

Oct. 23, 1984

The ARRL Packet-Radio Newsletter

TELEPORT STA GRANTED

The FCC has granted the joint AMSAT/ARRL request for a special temporary authorization (STA) permitting operation of automatic digital teleport stations. (For complete discussion of this request, and a list of stations involved, see Gateway issues 1 and 5.) The FCC "concluded that the automatic relaying of digital messages between amateur stations on earth and amateur satellites via intermediary stations (stations in teleport operation) may contribute to the advancement of the technical and communication aspects of the art of radio."

The following sections of the Amateur Radio Service rules have been waived for authorized stations:

> "97.126 (a) is waived to permit an amateur station engaged in teleport operations to retransmit automatically the radio signals of other amateur radio stations.
>
> "97.126 (b) is waived to permit a remotely-control led amateur station engaged in tcleport operations to communicate with stations which are not shown on such a station's network diagram.
>
> "97.79 (b) is waived to permit an amateur station engaged in teleport operations to be operated under automatic control without the presence of a control operator at a control point of the station; provided that devices are installed and procedures implemented to ensure compliance with the the Commission's rules at all times; and, provided further that upon notification by the Commission of improper operation of a teleport station under automatic control, such automatic control shall be discontinued immediately until all deficiencies have been corrected."

Authorized stations may operate satellite uplinks and downlinks on 144-146 MHz and 435-438 MHz. Terrestrial inputs and outputs may be on any amateur frequency at or above 50 MHz where digital communications is permitted.

This STA expires 180 days from October 18, 1984. Good luck to all stations involved.

PACKET RADIO TELECONFERENCE RADIO NET

"Packet Radio Overview and Prospective"wi11 be the subject of the December 2nd North American Teleconference Radio New (TRN). This net, heard on over 150 gateway stations (mostly VHF repeaters) across the U.S. and Canada and on the OSCAR 10 satellite, will explain what packet radio is, describe how to get started in it, point out the benefits to you, and outline the pitfalls to be avoided by the beginner and expert alike. The speakers on this TRN will be none other than Lyle Johnson, WA7GXD, and Harold Price, NK6K.

Lyle is President of the Tucson Amateur Packet Radio Corporation (TAPR) and was one of the primary developers of the TAPR terminal node contro 1 ler (TNC) hardware. For his work in developing the TAPR TNC, Lyle was awarded the 1984 Technical Exce 1 lence Award at Dayton. Looking to the future, Lyle is responsible for the processor design for the upcoming amateur packet satellite (PACSAT). He became active in packet radio in 1981, the pioneer days for this new technology.

Harold is a Director of TAPR and was on the team that designed the software for the TAPR TNC. He is also the AMSAT Project Manager for PACSAT. Harold is another packet-radio pioneer, having first become active in that technology in 1982.

Packet radio offers opportunities for both the traditional communicator and for the experimenter. Learn about packet radio from two of its leading developers by tuning into TRN, Sunday, December 2, 1984, at 6:00 PM CST (00002). For a complete list of gateway station locations and frequencies, write the TRN Manager, c/o Midway Amateur Radio Club, P.O. Box 1231, Kearney, NE 68847-1231 (S.A.S.E. please, Canada excepted). Those of you on CompuServe HAMNET can find this list in the XA4 data base.

From Midway Amateur Radio Club

GATEWAYS "DOWN UNDER."

On September 1, from 20002 to 00002, VK2BVD and ZL1AOX maintained a fully "connected" 1200-bit/s data link between Sydney, Australia, and Auckland, New Zealand, on the 20-meter band. Propagation conditions were excellent, and there was little multi-path over the 2650-km, one-hop path. Substantial files were transferred in each direction.

The following day, at the same time, an HF/VHF packet gateway was established using store-and-forward techniques with link-level acknowledgments. A total of 5 terminal node controllers (TNCs) and two computers were linked via 2-meter and 20-meter packet radio. Both ZL1AOX, in Auckland, and ZL3QL, in Cristchurch, were able to access the 2-meter local-area network (LAN) in Sydney.

The gateway was in operation again on September 3, with a VHF beam orientation problem solved. ZL1AOX was able to connect to a host computer at VK2ZRQ and interactively operate the machine for over an hour. Both the HF and the VHF ports of the VK2BVD gateway functioned as expected.

Other stations monitoring these activities included ZL3THJ, VK2AQG, VK2AYD, VK2XY, VK2ZXQ, and VK2KFJ. Via VK2BVD

TRANSLATION OF GATEWAY

As part of its continuing effort to promote digital communication modes, the Radio Club of Chile has been translating Gateway into Spanish. We received a photocopy of the Spanish edition. of the premier issue of Gateway, including an exce 1 lent trans lation of Jon Bloom's "Introduction to Packet Radio." If you need the Spanish edition of Gateway, contact

Radio Club De Chile
Departamento De Comunicaciones
Centro De Documentacion
Nataniel Cox 1054 - Santiago De Chile
CHILE

RF DESIGN TEAM

A few Motorola engineers, members of the Florida Amateur Digital Communications Association (FADCA), want to design radios for amateur digital operation around 900 MHz (there will soon be an amateur assignment in this band). This group, chaired by Tom Kneisel, K4GFG, is looking for individuals to help them define the interface between their radio hardware and network controller hardware that is under development elsewhere. These are competent and interested engineers, just looking for a little more coordination and information. If you can help, contact

Tom Kneisel, K4GFG
1600 S.W. 115th Ave.
Davie, FL 33325.

Via FADCA>BEACON

DEMONSTRATION IDEA

If you want to demonstrate amateur radio to the public, why not try a shopping mall? The South Brevard (Florida) Amateur Radio Club set up a booth in a local shopping mall, and generated considerable interest in amateur radio. Packet radio played a part in this demonstration, with forty-two third-party messages handled via packet. Twelve of the messages were delivered by the SOUTHNET packet-radio network, and the rest were passed along to the All Florida CW Net, part of the National Traffic System.

If you plan a packet demonstration (at a mall or anywhere else) remember to bring along plenty of backup equipment. The South Brevard group ran into strong electrical noise on their primary frequency, but since they had a couple of portable digipeaters in the mall parking lot, they were able to maintain solid connections to their message system. Via FADCA>BEACON

TSRAC PACKETS

In keeping with the feeling that Special Service Clubs (SSCs) should keep up with state-of-the-art techno logy, several member8 of the Triple-State8 Amateur Radio Club (TSRAC) have been experimenting with packet radio. Don Knollinger, WB8ZTV, and Jay, KD8GL, have been communicating via packet, and prompting their fellow club member8 to join the "digital communications wave of the future." TSRAC serves the area where Ohio, West Virginia, and Pennsylvania meet. Via TSRAC BET

NEWS FROM ROCHESTER

Fred Cupp, W2DUC, sent us a note giving the details of packet-radio activity in the Rochester, NY area. Fred says that there is now an active Rochester packet group with 7 stations on the air. Thursday night is packet activity night. Look for stations on the standard EASTNET frequency, 145.01 MHZ.

Via W2DUC

MT. ASCUTNEY PACKET CLUB

Another new packet club is the Mt. Ascutney Packet Radio Association (MAPRA), which serves the area around Mt. Ascutney, NH. The group's wide-coverage digipeater on 3,000-foot Mt. Ascutney is now a vital part of the growing EASTNET. For more information on MAPRA, contact

Car 1 Breuning , N1CB
54 Myrtle St.
Newport, NH 03773.

Via NEPRA PACKETEAR

NETWORKING IN CALIFORNIA

Packets from Sunnyvale (near San Fransisco) were received in Los Angeles and San Diego this week via several linked UHF repeaters, The path started on 146.58, went through two UHF hops (duplex audio repeaters), then out on 144.76, the alternate LA/SD packet channel. Packets from Oliver Barrett, KB6BA were heard by Harold Price, NK6K and Sumner Hansen, WB6YMH in Los Angeles, and by Mike Brock, WB6HHV in San Diego. More experiments are planned for the near future, including getting some packets up the other direction. From NK6K

UNIX ON EASTNET

Phil Karn, KA9Q, has connected an IBM PC-XT, running UNIX System III, to his packet-radio station. UNIX is a powerful,' multiuser, multitasking operating system, and Phil's computer can be accessed simultaneously via telephone, packet radio, and operator console. Programs current ly avai lab le include a sate 11 ite tracking program, mai 1 programs, and a PBBS-style message facility. In the future, Phil hopes to connect the system to a TNC capable of supporting several level-2 connections, allowing simultaneous

computer access to more than one packet-radio station. To use the system as a guest, simply type "guest" in response to the "login:" prompt, and type return when asked for a password.

From **KA9Q.**

MORE INFORMATION ON GRAPES

We have received some more information on the Georgia Radio Amateur Packet Enthusiast Society (GRAPES). The group is currently using a **146.13/.73** full-duplex digital repeater located on Sweat Mountain. The call of the repeater is **KD4NC-1.** A PBBS is in the works. GRAPES is running a packet net on 145.54 at 8 P.M., Sundays. The club's address is:

GRAPES
PO Box 223
Conyers, GA 30208.

Via **GRAPEVINE**

CAPRA MEETING

At its November meeting, the Chicago Area Packet Radio Association **(CAPRA)** will present a video tape on level-3 linking. The video tape was produced by members of the Central Iowa Technical Society **(CITS),** following the Iowa Section **Technical** Seminar. Lyle Johnson, president of TAPR, and Jeff Ward, ARRL computer engineer, were the featured speakers at that seminar, and the video tape should be interesting to those looking forward to a true amateur packet-radio network. The CAPRA meeting is at **2:00** PM, November 10, in the **Glenside** Public Library, **Glenside** Hts., IL.

Via **HAMNET**

SOFTWARE AVAILABLE

One of the reasons that packet radio has been able to grow very quickly is that many **technically-skilled** individuals have been willing to donate the fruits of their labors to the packet-radio community. One of the results of this nonprofit atmosphere is that packet bulletin board software and **TNC** software is available from several sources. Most of this software is in the public domain, and "sells" for the price of disks or PROMs. Whenever possible, Gateway will print notes on the availability of public-domain software for packet radio.

Lynn Taylor, **WB6UUT,** has written and debugged a very versatile packet bulletin board system **(PBBS)** program that runs on Apple II computers. Lynn's software was designed to be as self-maintaining as possible and to squeeze every drop of storage out of two disk drives. It runs under UCSD Pascal, and requires a TAPR TNC, a Thunderware Thunderclock, a CCS **7710** serial port and two disk drives. The software is believed to be robust and bug-free. (As with all free software, however, the user is asked not to complain if bugs are found.) Lynn is willing to give the software to anyone who sends him 3 disks and sufficient postage to return them. Send them to:

Lynn Taylor, WB6UUT
463 Myrtle Street
Laguna Beach, CA, 92651.

If you have a program that you would like to make available to the packet-radio community, send Gateway a description of the program.

PACKET-RADIO WITH THE XEROX 820

One of the decisions reached at the recent meeting of the ARRL Ad Hoc Committee on Amateur Radio Digital Communications was that the Xerox 820 computer would be used as a benchmark and a **testbed** for packet-radio networking software. While it is possible to send and receive packets using the serial I/O **(SIO)** on the 820, this job can be done more easily if a high-level data-link controller **(HDLC)** chip is added to the computer. In a recent issue of **FADCA>BEACON,** Howard Goldstein, **N2WX,** outlined his design for such a modification to the 820. After the Digital committee meeting, Howard and some members of TAPR went to work producing a kit for this **modification.** The kit, now called the "FAD board," is nearing completion; several prototype circuit boards will be finished by the time you read this. **After** Howard tests these first units, the circuit board will be made generally available. The FAD board is a "piggyback" board that replaces the Xerox 820 SIO with an 8530 HDLC chip, giving the user two RS-232-C or TTL HDLC ports. If you are interested in this circuit board, contact TAPR for ordering information.

From **G@WB9FLW**

ABSTRACTS

"How Good is Your Network Routing Protocol?," by Hsieh and Gitman, Data Communications, May, 1984. This article investigates one of the most important aspects of packet-network design -- packet routing. There are many routing procedures under consideration for amateur packet networks, and it is interesting to read about what the professionals do. The article addresses route generation, link cost, packet ordering, and packet-forwarding operation. Two charts detail the routing strategies used by ARPANET, TYMNET, SNA, DATAPAC, and TELENET, and the effects that the chosen routing strategies have on those networks.

"Implementing X.25 Communications Protocol," by Eric L. Beser, Microsystems, June, October and November, 1984. This series of articles is a tutorial on X.25, the CCITT packet protocol on which the AX.25 amateur standard is based. The first article reviews layered network architecture, and introduces state machines and their use in protocol implementation. The second article discusses the design of Pascal software to implement X.25 with the Intel HDLC chips. The third article will discuss implementation of X.25 using the Western Digital **WD2511.** If you are interested in designing a TNC, or just finding out how your TNC works, read this series of articles.

"RF Modems," by **Hatchett** and Howell, R. F. Design, September/October, 1984. This article is the first in a series of articles covering design and construction of RF modems. Several amateur groups

are currently working on such modems, so this is a timely article. From the ground up, this first article addresses the basics of RF modems, system design trade-offs, types of modulation, and FSK transmitters. It then finishes off with a discussion of integrated circuits that are useful in RF modem design. For the experimenter, there is a block diagram and schematic for a **1.5 Kbits/s** F SK modem.

HOW IS GATEWAY DOING?

As this issue of Gateway goes to print, there are more than 200 paying subscribers to the newsletter. Although I have no firm figures, I am sure that many other **packeteers** are receiving Gateway via electronic mail services or amateur packet-radio networks. I would like to thank you for your support, while reminding you that Gateway would like to hear from you about packet activity in your **area.**

Jeff Ward, **K8KA**

REPRODUCTION OF GATEWAY MATERIAL

Material may be excerpted from Gateway without prior permission, provided that the original contributor is credited and Gateway is identified as the **source.**

Gateway: The ARRL Packet Radio Newsletter is published by the

American Radio Relay League
225 Main Street
Newington, CT 06111

Jeffrey W. Ward, **K8KA**
Editor

Larry E . Price, **W4RA**
President

David Sumner, **K1ZZ**
General Manager

Subscriptions are available to ARRL members and nonmembers, at the following rates:

For 25 issues of Gateway:
- In the U.S.
  - ARRL Member $6 .00
  - Nonmember $9.00
- In U.S., Canada, and Mexico by First Class Mail
  - ARRL Member $11.00
  - Nonmember **$14.00**
- Elsewhere by Airmail
  - ARRL Member **$14.00**
  - Nonmember **$17.00.**

Prices are subject to change without notice. Remittance must be in U.S. funds, and checks must be drawn on a bank in the U.S.

Gateway: The ARRL
Packet Radio Newsletter
American Radio Relay League
225 Main **St.**
Newington, CT 06111

Nonprofit Organization
U.S. Postage
PAID
Hartford, Conn.
Permit No. 2929

Gateway

Vol. 1, No. 7
Nov. 7, 1984

The ARRL Packet-Radio Newsletter

ARRL BOARD MAKES PACKET-RADIO DECISIONS

The ARRL Board of Directors met in Hartford, Connecticut, on October 25 and 26. Two matters of interest to packet-radio operators were discussed.

First, the Board approved the AX.25 level-2 protocol specification submitted by the ARRL Ad Hoc Committee on Amateur Radio Digital Communications .

Second, the Board approved, "in substance," a draft of a petition requesting amendment of amateur regulations to permit automatic control of digital stations operating above 30 MHz. This petition will now be made formal by the ARRL legal counsel and then submitted to the FCC for considerat ion. This rules change, should it be enacted, will clear up any doubt as to where packet-radio bulletin boards and other network servers fit into amateur regulations.

CALL FOR PAPERS

The American Radio Relay League will hold its Fourth Amateur Radio Computer Networking Conference on March 30, 1985, in San Francisco, CA. The conference will be in cooperation with the West Coast Computer Faire being held March 30 through April 2.

Technical papers are invited on all aspects of amateur packet radio and other forms of Amateur Radio digital communications via terrestrial, ionospheric, meteor-scatter and satellite media including AMSAT-OSCAR 10 and PACSAT. Topics may include network and system architecture, proposed standards, hardware, software, protocols, modulation and encoding schemes, applications, and practical experience.

The deadline for receipt of camera-ready papers is March 1, 1985. All papers should be mailed to Mar ian S. Anderson, WB1FSB, American Radio Relay League, 225 Main Street, Newington, CT 06111. If you plan to present a paper, please request an author's kit and identify the title of your paper immediately. Proceedings will be sold at the conference and by mail from ARRL Hq.

From **W4RI**

CHERRYVILLE REPEATER ASSOCIATION

We received the following letter from Charlie Kosman, WB2NQV:

"I would like to call to your attention an active packet group in New Jersey. The Cherryville Repeater Association has been in the forefront of packet operation starting with the original Vancouver boards and even starting a repeater interlink for voice discussion on packet protocol with other groups around the country. We are presently on the New Jersey 220 Repeater Network with AX.25 Ashby boards on 224.12, Mt. Kipp; 224.52, Philadelphia; 220.54 Delaware Water Gap full time and can be accessed also from 223.78, Warrenville, NJ; 224.50, Newton, NJ; and 224.48 MHz, Montclair, NJ.

"We can also connect to any two-meter frequency or the telephone line for access to other networks.

"A PBBS (packet bulletin board system) is available , and plans call for a main-frame PBBS and a teleport via OSCAR 10 soon.

"We are even using packet for public service. In a rescue drill coinciding with SET (Simulated Emergency Test) weekend in October, relaying data on 80 victims to area hospitals from the disaster scene using portable packet terminals.

"Our slogan sums it up: 'Amateur Radio at its finest: community service through communication.'

"For more information call the club at (201)788-4080 or write:

Cherryville Repeater Association, Inc.
Box 308
Quakertown, NJ 08868."

Bill Ashby, K2TKN, is also a member of the Cherryville Repeater Association. He tells us that several stations on 220 MHz in New Jersey are using the V.2 protocol written by Doug Lockhart, VE7APU. While stations using V.2 can't communicate with stations using AX.25, the experience of the group in New Jersey is that the two protocols can be used on the same frequency without any problems.

Another interesting thing about this group is that they are using full-duplex repeaters shared with voice and RTTY users.

From **K2TKN** and **WB2NQV.**

TRAFFIC ON SOUTHNET

"There are 5 or 6 stations on SOUTHNET [Florida] that have been handling traffic since February. Messages are passed in radiogram format, although there has been some disagreement on what is the best message form, especially for messages that are handled only on packet.

"The big problem here is gaining formal. recognition by an EC or STM as an NTS net. I'm not sure if the basis of the difficulty is that these officials don't understand packet or the fact that there is no specific net time (QND) for these packet nets. I'd like to hear from anyone who has gained or has suggestions for getting this designation.

"The SOAPnet (SOUTHNET Amateur Packet traffic net) has outlets to the QFN Florida all-sections CW net (Cycle 4) and a couple of local ARES nets in the southern part of the state.

"The big event of the SOAPnet was a traffic fair at a mall which used packet to relay the cache of traffic (about 40 messages) to a text editor. From there, one-quarter of the messages were cleared via packet channels alone; the balance was taken to QFN."

From Howard Goldstein, N2WX.

UNIX NOTE

In Gateway issue 6, we noted that KA9Q has brought a UNIX system up on EASTNET. Those who sign onto the system (using the directions in Gateway) should type "info" when they see the "ka9q$" prompt. This will generate a brief introduction to what UNIX is all about and some instructions on how to do a few simple things such as send and receive mail.

Via KA9Q.

DIGIPEATER AS REMOTE TNC

Ted Huf, K4NTA, operates the wide-area digipeater K4NTA-1 in Stuart, FL. The digipeater is located at Ted's place of work, which happens to be a CATV company. The antenna is on the main CATV receiving tower at 300', and it is fed with 7/8" hardline. The TNC is a TAPR Beta board, and the radio is a GE Master. All of this is not too unusual, but Ted also uses the TNC by remote control from his home. The TNC is connected to a dumb terminal in Ted's shack via two 4800-bit/s CATV RF modems made for data communications on cable systems. The remote control has worked out very well, and since there are not many mountains in Florida, Ted has one of the best sites in his area.

Ted reports further: "We now have wide-area-coverage digipeaters in Jacksonville, Melbourne, Stuart and West Palm Beach, Florida. W1BEL's home station provides the link into the Tampa area on the west coast of Florida. Digipeaters are planned for Okeechobee, Miami, Orlando, Tampa Lake Wales and Ocala. There are now two PBBS stations on the air: K4NTA in Stuart and WD4LHF in West Palm Beach. Connections are routinely made with all areas that have packet activity. I estimate that there are around 100 stations on in Florida now. We are planning such things as PBBS linking for message exchange and would like to hear from others working on this."

Prom K4NTA.

COORDINATION IN MINNESOTA

The Minnesota Repeater Council, at their October 27 meeting, voted to set aside 145.0 MHz to 145.1 MHz exlusively for single-channel digital communications. This includes single-channel repeaters being used for packet radio. The allocation is divided into five 20-kHz channels, with the first channel centered on 145.01 MHz.

From WØTN

TELEPORT NEWS

As a first phase of the new AMSAT/ARRL Teleport STA, a unique packet-radio test was carried out on October 28. An automatic packet-radio bulletin board system (PBBS), operated by Tom Clark (W3IWI), was placed in experimental operation on the AMSAT-OSCAR-10 satellite. The W3IWI PBBS was successfully used by several amateurs across the U.S. and Canada. Earlier this year, successful PBBS tests through the same satellite had involved gateway links to existing terrestrial PBBSs in California.

The following stations participated in the test and successfully logged onto the W3IWI PBBS:

Randy Smith, VE1PAC/VE6, Medley, Alberta;
Wes Morris, K7PYK, Scottsdale, AZ;
Mac Jordan, W4DAQ, Demopolis, AL;
Bob Diersing, N5AHD, Corpus Christi, TX.

A number of other stations were monitoring and reported good copy.

Both VE1PAC and K7PYK sent and received several messages and files during their connections with the PBBS. K7PYK maintained contact for about an hour and managed to acquire about 50 kbytes of documentation from W3IWI. W3IWI and VE1PAC tested full-duplex PBBS operation through the satellite and achieved sustained data throughput of about 1 Kbit/s, despite the 0.25-second round-trip propagation time to the satellite.

The packet-radio hardware used at W3IWI is normally used as a local area PBBS, serving the Baltimore-Washington network. The PBBS uses a Xerox 820 computer wit8 software by Hank Oredson, WORLI. The terminal-node controller (TNC) used is a TAPR beta-test model. All tests were conducted at 1200 bit/s, 1000-Hz shift AFSK.

Both N5AHD and W3IWI are among the 24 stations authorized by STA to operate unattended, automatic teleport stations.

Packet operation on AMSAT-OSCAR-10 uses a downlink near 145.835 MHz, about 5 kHz up from the nominal AMICON digital network frequency of 145.830. (Operation was moved up to avoid sensitivity degradation experienced near the edge of the satellite passband. For packet radio, every dB counts.)

From W3IWI

AUTOMATIC MESSAGE FORWARDING

Don Haney, KA1T is working on a PBBS devoted to NTS-type message handling, and he hopes to have

the system up by December. Don is looking for other stations in New England that are interested in working on automatic FBBS-to-PBBS message forwarding. This type of store-and-forward operation will soon become very important to the growth of amateur packet radio. Interested parties should contact

Don Haney
Rd 1 Box 237
Harvard, MA 01451.

Prom **KA1T**

FLORIDA PACKET CENSUS

Howard Goldstein, **N2WX,** has compiled a list of stations active on packet radio in Florida. He lists 74 individual operators with a total of 77 stations on the air. Five of the stations are wide-coverage digipeaters, and three stations are in bullet in-board service.

Via **DR NET**

TEXAS PACKET ACTIVITY

David Cheek, **WA5MWD,** from the Dallas, TX area, sends along the following:

"I've been on packet since August, 1983, first with a Vancouver TNC and now with a TAPR board. My main interest is in developing message handling systems for emergency service uses. This makes linking, high-level message relay, portable operations, and meteor scatter the specific areas that I expect to work in. The Texas VHF-FM Society appointed me to chair a group that will decide what frequencies should be used for packet radio in Texas."

Via **DR** NET

ARRL PACKET ACTIVITY

While Gateway has been concerned with reporting on packet-radio activity throughout the world, we have not provided an overview of packet activity in our own area, central Connecticut. In fact, the W1AW-4 packet bulletin board system (PBBS) and the W1AW-5 digipeater are at the center of a growing packet community in Connecticut and southern Massachusetts.

The W1AW-4 PBBS is located at ARRL Headquarters. The PBBS software was written by Jon Bloom, KE3Z, to run on a Xerox 820 computer connected to a TAPR TNC. The TNC that is currently being used is a TAPR beta-test model belonging to Paul Rinaldo, W4RI. RF is provided by an old Genave Z-meter rig. Because the W1AW bulletins and code practice are transmitted only a few hundred feet from the PBBS antenna, the PBBS does experience some reception problems during W1AW operating hours. There have been 300 messages left on the PBBS during its six months of sporadic operation.

The W1AW-5 digipeater is located at the home of Dave Sumner, K1ZZ, General Manager of the ARRL. Originally, the digipeater used a VADCG TNC (running AX.25 protocol) and a VHF Engineering repeater. Due to repeated, mysterious failures of this system, it has been replaced by an AEA PKT-1 TNC and a Heathkit 2-meter transceiver. Dave has put the W1AW-5 antenna on his 125-foot tower, and the repeater can be accessed reliably from throughout central Connect icut and south-central Massachusetts.

When W1AW-5 is operating, stations in Connecticut can use several digipeaters in Massachusetts to connect with stations in the Boston-area packet network. Proposed digipeaters in eastern New York would connect W1AW-5 to New Jersey, the southern portion of **EASTNET,** and beyond.

W1AW is one of the stations authorized by STA to operate as an unattended, automatic teleport stat ion. The RF hardware for satellite operation has been installed at the W1AW laboratory station, and teleport experiments will be started soon.

We would like to thank Advanced Electronic Applications for donating a Z-meter antenna for W1AW-5 and Mirage for donating an amplifier for the digipeater.

From K8KA

ERROR IN GATEWAY **ISSUE** 6

In the abstract of "RF Modems" by Hatchett and Howell, we mentioned a block diagram and schematic for a 1.5-kbit/s RF modem. In fact, the design is for a 1.5-Mbit/s modem. That makes it a bit more interesting.

Via **K1HOP**

ACTIVITY IN **UTAH**

Ron Todd, K3FR, ARRL Section Manager in Utah, reports that there is some packet-radio activity in the in and around Utah's Salt Lake valley. There are already 3 or 4 stations on the air, and others under construction. If the proper digipeater sites can be located, Utah could become an important link between the Pike's Peak area (covered by the Rocky Mountain Packet Radio Association) and stations in the Pacific northwest. Because of this, Ron is particularly interested in high-level linking.

via **K3FR**

AMSAT MEETING

AMSAT is holding its Second Annual Amateur Radio Satellite Symposium on November 10, in Los Angeles, CA. Of interest to Gateway readers will be "Computers and the Satellites," presented by Bob Diersing (N5AHD) and "PACSAT Forum" with Harold Price (NK6K), Wally Lindstruth (WA6JPR), Rick Fleeter (WA8VGK), and Phil Karn (KA9Q). The next issue of Gateway will report on these meetings.

DIGITAL **COMMITTEE MEETING**

The ARRL Ad Hoc Committee on Amateur Radio Digital Communications will meet in conjunction with the AMSAT Symposium on November 10. At this meeting, groups will report on the progress made toward

implementation of level-3 protocols, the state of development of high-speed modems, and availability of HDLC piggyback boards for the Xerox 820. Rumour has it that a 9.6 Kbit/s FSK modem for 220 MHz, designed by Steve Goode, K9TD, will be demonstrated.

DATAGRAMS AND VIRTUAL CIRCUITS

If you would like to learn more about the various approaches to packet networking, and two of the approaches to it being considered by amateur packet groups, you might be interested in Distributed Systems -- Architecture and Implementation, Lampson, Paul and Siegert, editors, Springer Verlag, 1981. Of particular interest is section 6.6 which discusses, in some detail, datagrams and virtual circuits.

From KA9Q.

REPRODUCTION OF GATEWAY MATERIAL

Material may be exerpted from Gateway without prior permission, provided that the original contributor is credited and Gateway is identified as the source.

Gateway: The ARRL Packet Radio Newsletter is published by the

American Radio Relay League
225 Main Street
Newington, CT 06111

Jeffrey W. Ward, K8KA
Editor

Larry E. Price, W4RA
President

David Sumner, K1ZZ
Genera 1 Manager

Subscriptions are available to ARRL members and nonmembers, at the following rates:

For 25 issues of Gateway:
- In the U.S.
  - ARRL Member $6 .OO
  - Nonmember $9 .OO
- In U.S., Canada, and Mexico by First Class Mail
  - ARRL Member $11.00
  - Nonmember $14.00
- Elsewhere by Airmail
  - ARRL Member $14.00
  - Nonmember $17.00.

Prices are subject to change without notice. Remittance must be in U.S. funds, and checks must be drawn on a bank in the U.S.

Gateway: The ARRL
Packet Radio Newsletter
American Radio Relay League
225 Main St.
Newington, CT 06111

Nonprofit Organization
U.S. Postage
PAID
Hartford, Conn.
Permit No. 2929

Gateway

Vol. 1, No. 8
Nov. 20, 1984

The ARRL Packet-Radio Newsletter

ARRL AND PACKET RADIO

As mentioned in the last issue of Gateway, the ARRL Board of Directors, at its recent meeting, took several actions that affect packet radio. (See Board Minutes 11, 30, 36-38, 70, 98-99 in December, 1984 QST.) Deserving special attention are Minutes 98 and 99:

"98) ... it was unanimously VOTED that the Ad Hoc Committee on Amateur Radio Digital Communications, packet radio clubs, and packet radio experimenters are commended for their work to date, and in particular, for their reaching agreement on the AX.25 link-layer protocol.

"99) ... it was unanimously VOTED that the Proposal for Packet-Radio Development Program...be adopted."

These two Board actions recognize that packet radio has come a long way in a short time, and that it still has a long way to go.

The "Proposal for Packet-Radio Development Program" mentioned in Minute 99 is a document authored by Paul Rinaldo, W4RI, that details recent growth in packet radio, the goals of packet-radio experimenters, and actions that the ARRL might take to assist packet-radio development. While it does not attempt to allocate funds or personnel to packet-radio, it ends with the following list of recommended actions:

1. Board Actions

   a. Approve the AX.25 link-layer protocol.

   b. Support the Digital Committee.

   c. Approve ARRL laboratory support of packet radio.

   d. Approve San Francisco as the site for the Fourth ARRL Amateur Radio Computer Networking Conference.

   e. Encourage more Board Members to get on the air with packet.

2. Headquarters Actions

   a. Publish the AX.25 Amateur Packet-Radio Link-Layer Protocol document.

   b. Develop a nontechnical flyer on packet radio for answering mail inquiries and as a handout for field use.

   c. Publish more packet-radio articles in QST and do club profiles on clubs contributing to packet radio.

   d. Work with the Connecticut Section Leadership Officials to develop a demonstration "model section" with a variety of packet-radio services.

   e. Continue ARRL lab technical support of packet-radio experimenters, particularly in preparing descriptions of designs for League publications.

   f. Configure W1AW for packet-radio teleport operation and conduct tests with other teleports under a special temporary authority granted by the FCC.

   g. Produce a slide show or video tape on packet radio.

3. Ad Hoc Committee Actions

   a. Coordinate and support experiments leading to an early agreement on network and transport protocols.

   b. Develop a message protocol capable of supporting Amateur Radio emergency communications, NTS traffic, inter-PBBS exchanges, and other electronic-mail applications.

   c. Work towards developing standards for other protocol layers.

   d. Encourage development of higher-speed packet-radio systems.

   e. Investigate ways of reducing the cost of packet radio to amateurs.

Obviously, not all of these items will be acted on immediately, and some may give way to more fruitful or critical projects. However, this program does indicate that the ARRL Board recognizes the potential of packet radio to improve Amateur Radio, making it a more interesting, advanced hobby and a more effective tool for public service.

ED

ARRL DIGITAL COMMITTEE MEETING

The ARRL Ad Hoc Committee on Amateur Radio Digital Communication held a meeting in Los Angeles, on November 11. Highlights from the Meeting Minutes follow.

The ARRL VHF/UHF Advisory Committee (VUAC), and the Southern California Repeater and Remote Base Association (SCRRBA) have reached an agreement on a national band plan for the 23-cm band. The Digital Committee reaffirmed its stand that the digital communications frequencies provided in the now band plan would be adequate for the development of packet radio.

Tucson Amateur Packet Radio (TAPR) has completed prototypes of the six-chip packet I/O add-on for the Xerox 820 computer. These add-ons should be available from TAPR at the end of November. [Please wait for an official TAPR announcement before attempting to order this kit. -- Ed]

The Committee discussed how best to present information on packet-radio repeaters in the ARRL Repeater Directory. The committee will recommend that digital repeaters be listed in the main body of the directory as well as in a special section devoted to digital repeaters. This would give visibility to digital operations, perhaps alerting amateurs to digital activity in their area. The Committee also recommended that secondary-stat ion identifiers (SSIDs) be added to the call signs of packet repeaters listed. It was further suggested that the Repeater Directory include a list of frequencies used for packet operation, and that a system be developed for identifying stations providing special services (such as gateways, teleports and bulletin boards).

Following up on an earlier discussion, the Committee agreed to encourage TNC developers to cooperate in standardizing the interface to user terminals. Doug Lockhart, VE7APU, will be the Committee's clearing house for this matter. Such a standard interface will be similar to that presented in CCITT Recommendations X.3 and X.28 and will make it much easier for users to operate TNCs made by different manufacturers.

The Digital Committee discussed several other matters, including transceiver turnaround delay, message forwarding, and the problems of mountain-top packet installations. As various projects undertaken by the Committee mature, they will be addressed in Gateway.

Via W4RI

AUTOMATIC CONTROL

It was reported in the last issue of Gateway that the ARRL was preparing a pet it ion for rule making asking the FCC to allow automatic control of digital communications above 30 MHz. Excerpts from the petition, filed on November 14, follow.

"The American Radio Relay League...requests that the Commission. ..permit automatic control of digital communications on all amateur frequencies above 30 MHz, provided that certain safeguards are incorporated in the amateur station as control functions...

"There are now...provisions for use of digital communications in the Amateur Radio Service. ..These types of digital communications, especially since the advent of request-repeat radioteleprinter systems such as AMTOR, can be done with a great deal of reliability without a control operator present. Amateur stations, utilizing present microprocessor and computer technology now routinely present at amateur stations, can automatically transmit and receive digital communications, verify receipt of messages, and respond when inquired of. The use of Computer Based Message Systems is a new aspect of amateur communications which should be encouraged, consistent with establishment of standards for good amateur operating practice. digital communications has progressed to the point that automatic control of digital communications is both feasible and necessary to facititate further development of such experimentation.

"...automat ic operation would be subject to the inclusion of adequate circuitry to assure (1) the detection of transmitter malfunction and, upon detect ion thereof, automatic transmitter shutoff; (2) the capacity to prevent transmission of improper message traffic, such as business communications; (3) means to prevent transmission while the channel is occupied; and (4) compliance with all other applicable technical and operational standards for amateur radio stations.

"Of course, the authority to utilize automatic control of digital communication would not alter the primary responsibility of the station licensee for proper station operation. The League Board of Directors has adopted interim standards for good amateur practice for digital communications, and specifically for establishment and operation of computer-based message systems ... [See October, 1984 QST, "Operating News"--Ed.] The promulgation of such guidelines will assist in assuring responsible use of the automatic control sought by this Petition.

"The Commission has twice recently emphasized its objective of encouraging new technologies' in the Amateur Service, balanced against assuring enforcement capability. In this context, no enforcement problems are created by the proposed rule change. Yet, the added authority would greatly facilitate and encourage amateur digital experimentat ion and increase communication effectiveness.

"Given the heavy use of high frequencies [below 30 MHz], it is believed that manual control for digital communications is more appropriate [on those frequenc ies]."

Remember, this petition for rule making is just the first step in the long process of requesting and gaining a rules change. Gateway will keep you informed on the progress of this pet it ion.

ST. LOUIS DIGIPEATER

On Saturday, October 10, St. Louis Area Packet Radio (SLAPR) installed its first permanent dig ipeater , KØPFX-1. The digipeater is located in downtown St. Louis on the TV channel 30 tower, at the 400-ft level. The antenna (a Ringo Ranger) is fed with 400 feet of hardline. The TNC is a TAPR beta board that has been upgraded to perform like a TAPR kit TNC, The RF end of the installation is an IC-22S running 9 watts. Several St. Louis-area amateurs donated the equipment and packaged it in a cabinet purchased by SLAPR. Wiring and installation of the loaned equipment was undertaken by Mel Whitten, KØPFX.

Although the feedline is long and the rig is not running high power, signal reports from the first evening of operation were promising. Packet stations as far away as Greenville, Illinois were able to copy. This is a 40-mile path -- not bad for the flatlands of the midwest!

From WB9FLW

VANCOUVER DIGIPEATER

The Vancouver Amateur Digital Communications Group (VADCG), one of the groups that pioneered amateur packet radio, has installed a 145.65-MHz wide-coverage digipeater on Burnaby Mountain. The equipment is a VADCG TNC with repeater and beacon software, an IC-22S transceiver, and a multi-element beam aimed south (toward the U.S.). Coverage is good into Bellingham, Washington, and reports have been received from as far south as Edmonds, Washington. Local coverage includes greater-Vancouver and the Fraser Valley. There are about a dozen Canadian stations active on the digipeater, and there are several active stations in Bellingham. Others are still in the process of bringing up TNC boards and modifying VADCG TNCs to support the V.2 protocol. Many stations have Xerox 820 boards providing network services, and WA7PIX has had his Xenix system on line. Packet operators visiting the Vancouver area are encouraged to use the machine.

From VE7DPM

SEATTLE, WA PACKET CLUB

A packet-radio club called the Northwest Amateur Packet Radio Association (NAPRA) was formed in Seattle, WA on November 13. The officers of the club are Dennis Goodwin, KB7DZ; John Gates, N7BTI; John Hays, KD7UW; Bud Churchward, WB7FHC; and Tom Hogan, WB7DCH. The group has about 30 members. The Seattle area supports two hilltop digipeaters (on 147.6 MHz and 224.56 MHz) and two packet radio bulletin board systems (PBBSs), with coverage from Olympia to Bellingham. Club meetings will be on the second Saturday of each month, and the club holds a voice net on 145.33, Thursday from 2100 to 2200 local time. Plans are to link the Seattle area to British Columbia, eastern Washington, and Portland, Oregon.

Via N7TBI

ACTIVITY IN VIRGINIA

We received the following letter from Peter Lascell, W4WWQ.

"Packet is coming to central and southwest Virginia after WD4OLV recently moved into the area. Rick has presented a couple of club programs in the area, and the ball is rolling.

"About a half-dozen stations are actively assembling equipment in the Roanoke, VA area, and there are plans to install a 145.01-MHz digipeater atop Poor Mountain, near Roanoke. Poor Mt. is about 4,000 ft high, and is the current site of the WB4QOJ voice repeater. The RF equipment is ready, and we are just waiting for the GLB PK-1 to make it fly. The earliest date for complete installation is the Thanksgiving Day weekend. This repeater should provide coverage down to the Florence, SC packet repeater as well as up the Shenandoah valley to EASTNET near Washington, D.C. Users in Greensboro, NC, Blacksburg, VA, and Lynchburg, VA, will have easy access to the Poor Mountain digipeater. More distant stations, from areas like Raleigh-Durham and southeastern West Virginia should be able to access the machine if they have good outdoor antennas.

"There are at least five stations in Lynchburg, VA with TNCs in their shacks, although some of the TNCs are still in kit form. There are basically two levels of computer interest here -- those with Vic 20 or C 64 computers who still play games when the propagation is bad, and the CP/M crowd with Big Board and Xerox 820 computers.

"As activity builds, we will evaluate the merits of another repeater on the east side of the Blue Ridge. The primary intent of such a machine would be to link the western part of the Commonwealth with Richmond, providing access to the State Office of Emergency Services. (We are very much public service oriented.)"

From W4WWQ

NEWS FROM GLB

From the Packet Software Approach Newsletter, we hear that GLB electronics expects to have a "PK2" in a few months. The use of a Zilog Serial I/O (SIO) chip should allow the PK2 to run much faster than the current GLB PK1, which uses no SIO.

Via W4UCH

PACKET IN ONTARIO

The Hamilton Amateur Packet Network (HAPN) now has two digipeaters on the air: VE3PKO on 145.710 MHz, and VE3PKT on 145.650 MHz. One of these machines is in Toronto, and one of them is just south of Toronto. Most of the stations on HAPN are using Vancouver TNCs modified to run AX.25 protocol. Stewart Beal, VE3MWM, even has AX.25 and the Vancouver protocol running on his VADCG TNC. If you are in the area, HAPN runs a voice net on two meters--147.735/147.135 MHz, Monday night.

via W4UCH

PACKET READING

Looking for something to read about packet-switched data networks? The following list came to us via Compuserve's HAMNET:

Communications Networks for Computers, by Davies and Barber.

Computer Networks and Their Protocols, by Davies, Barber, Price, and Solomonides.

Data Communications, a User's Guide, by Kenneth Sherman.

Packet Switching, by Roy Rosner.

Telecommunications and the Computer, by James

Martin.

Telecommunications System Engineering, by Freeman.

From **W2JUP**

WHERE TO FIND **THEM**

You probably won't find these books in your local book store or even at your public library. The best place to find books and magazines mentioned in Gateway is at the library or book store of a college of engineering. There is usually some way for people who are not students or staff members to check out books -- just don't take "no" for an answer.

From **K8KA**

REPRODUCTION OF GATEWAY MATERIAL

Material may be exerpted from **Gateway** without prior permission, provided that the original contributor is credited and Gateway is identified as the source.

Gateway: The ARRL Packet Radio Newsletter is published **by the**

American Radio Relay League
225 Main Street
Newington, CT 06111

Jeffrey W. Ward, **K8KA**
Editor

Larry E. Price, **W4RA**
President

David Sumner, **K1ZZ**
General Manager

Subscriptions are available to ARRL members and nonmembers, at the following rates:

For 25 issues of Gateway:
- In the U.S.
  - ARRL Member $6.00
  - Nonmember $9.00
- In U.S., Canada, and Mexico by First Class Mail
  - ARRL Member $11.00
  - Nonmember $16.00
- Elsewhere by Airmail
  - ARRL Member $14.00
  - Nonmember **$17.00.**

Prices are subject to change without notice. Remittance must be in U.S. funds, and checks must be drawn on a bank in the U.S.

Gateway: The **ARRL**
Packet Radio Newsletter
American Radio Relay League
225 Main St.
Newington, CT 06111

Nonprofit Organization
U.S. Postage
PAID
Hartford, **Conn.**
Permit No. 2929

Gateway

Vol. 1, No. 9
Dec. 4, 1984

The ARRL **Packet-Radio** Newsletter

AUTOMATIC FORWARDING

From the user's point of view, the most interesting packet-radio development of the past few weeks is the implementation, by Hank Oredson, WØRLI, of automat ic message forwarding. Hank is the author of The MailBox, a packet-radio bulletin-board system (PBBS) that runs on the Xerox 82Ø computer. The newest feature of The MailBox is automatic store-and-forward message hand 1 ing, which allows a message left on a MailBox to be passed from one MailBox to another until it reaches its intended recipient. our experience with the system, between Hank, in Boston, and W1AW-4 in Newington, CT, has been f avorab le.

Hank says, "The distributed message store-and-forward system is growing very rapidly. h e Boston area, K1BC, WØRLI, WB2OSZ, and KA1T are all using auto-forwarding. In Arizona, K7PYK is available, linked to WØRLI via HF. Other EASTNET MailBoxes now on the air, or soon to be on the air, include W3IWI and KS3Q in the Washington, D.C. area, WB2MNF in New Jersey, and W1AW-4 in New ing ton. At least nine other stations should be coming up in the future. The network, both real-time and store-and-forward, is growing fast."

While the easiest way to get into the store-and-forward network is to use Hank's MailBox software, experience at W1AW has shown that other PBBS software can be modified to work with the MailBox. Hank would like to get in touch with anyone working on store-and-forward techniques. [See "The MailBox" later in this issue of Gateway for more information on Hank's software. -- Ed.]

Ed.

TELECONFERENCE RADIO NET

Packet radio was the topic of the December 2nd Teleconference Radio Net (TRN), which featured Lyle Johnson, WA7GXD, and Harold Price, NK6K. This net, heard on many repeaters throughout North America, should generate a lot of new interest in packet radio.

Lyle and Harold addressed the basics of packet radio, the state of packet radio, packet-satellite projects, and the future of packet radio. Questions from the audience showed increasing awareness of and interest in packet radio.

Congratulations to Lyle and Harold for presenting a clear and interesting introduction and overview of packet radio. Congratulations also to the Midway Amateur Radio Club for its first TRN; the net was well controlled, and the audio was beautiful.

Ed.

PACKET RADIO VIDEO TAPES

The Central Iowa Technical Society (CITS) was privileged to host the first ARRL Iowa Section Technical Seminar on September 22nd, 1984. The guest speakers at that convention were Lyle Johnson, President of Tucson Amateur Packet Radio (TAPR), and Jeff Ward, K8KA, ARRL laboratory computer engineer and editor of Gateway. The weekend before the Seminar, Lyle and Jeff had attended a meeting of the ARRL Ad Hoc Committee on Amateur Radio Digital Communications, and they were well prepared to present the latest ideas and agreements relating to packet-radio networking.

CITS took the opportunity to video tape Lyle's discussion of the concepts of networking and, with the addition of a segment about the Iowa networking project, they have produced a video tape called "Amateur Packet-Radio Networking."

Lyle's d iscuss ion includes a brief review of packet-radio basics, a discussion of the use and limits of level-2 digipeaters, a comparative review of datagrams and virtual circuits, and speculation on potential of packet-radio networking to provide continuous, highly reliable, error-free inter-city linking.

CITS has followed Lyle's discussion with a description of their own networking project, which includes a "link-management unit" (LMU) based on surplus Televideo 8Ø6 microcomputer boards and 22Ø-MHz RF modems. This description is in the form of a panel discussion including John Maurer, NAØS, Dave Huffman, KAØDNB, and Ralph Wallio, WØRPK.

The video tape is ready for distribution and may be ordered in either VHS or BETA format. Please send $25 (in check made out to Central Iowa Technical Society) to:

Allen Johnson, WBØOEU
c/o CONTROLTRACK
2212 Ingersoll Avenue
Des Moines, IA 5Ø312.

If you are just getting interested in packet radio, perhaps as a result of the December 2 TRN, CITS's first video production, "Introduction to Packet Rad io" is still available through the same address, for the same cost. Funds remaining after CITS's overhead is paid will be used to continue

development of packet-radio networking.

From WØRPK

RATS ACTIVITY

The following summary of the activities of the Radio Amateur Telecommunications Society (RATS) was sent to us by Gordon Beattie, N2DSY.

"The new repeater is up and running in a revised form. It sports an Ashby TNC, a Bell 2Ø2T modem, a dual-function watchdog timer, and an ICOM IC-25A. The new site, in Little Falls, New Jersey, improves coverage of the New York metro area. The package is now housed in an all-weather cabinet provided by Mike Friedman, WB2WNX. This will eliminate operational problems encountered during each of the past winters. The dual-timer circuit provides a timeout for the transmitter, and a reset timer in case the repeater "goes to sleep." This circuit has already paid for itself in saved gasoline.

"Our bulletin board will be on the air in the next few weeks, providing access to news and reference data for both advanced and beginning packeteers. We will be running a modified RBBS on a Big Board computer with 75Ø kbytes of disk storage. We are also investigating a link to the New Jersey Institute of Technology's Electronic Information Exchange System. This system provides information on many subjects, including computers, sociology, language, mathematics and leisure. This system is also host to the Digital Radio Net (DR NET) group that is coordinating packet development efforts here in North America.

"We've also been busy in the protocol development area. We have a draft recommendation for an international numbering plan and another in the works for a national numbering plan. The international plan closely follows the CCITT x.121. This is a particularly interesting recommendation because it allows for data users to call telephone and telex users. An amateur implementation of this addressing scheme would simplify connections between amateur and commercial net works. This would greatly enhance amateur network operations, provided that proper control and third-party traffic regulations could be observed.

"The national plan to be proposed will be based on the North American Numbering Plan used in the telephone network. Our goal was to develop an easily determined numbering scheme for use in the amateur network, and we couldn't think of a more reliable directory service than the one maintained by AT&T and others!"

[Numbering will be necessary in an amateur packet network because amateur call signs are not geographically specific enough to be used in determining packet routing. Several numbering schemes are under consideration. -- Ed.]

For more information about RATS, contact:

RATS
c/o J. Gordon Beattie, Jr., N2DSY
2Ø6 North Vivyen St.
Bergenfield, NJ Ø7621.

From N2DSY

EASTNET NEWS

Tom Clark, W3IWI, reports on the central division of the East Coast Packet Network, EASTNET:

"On sunday, November 25, about a dozen active packeteers from Maryland (Baltimore and Washington D.C.), southern New Jersey (Camden) and southeastern Pennsylvania (Harrisburg) met at a rest stop on Interstate 95 near the MD/DE border (approximately equidistant from the three centers) for the purpose of coordinating EASTNET activities.

"One of the major topics at the meeting was the availability of digipeaters. The critical "hub" of intercity communications is the WB4APR-6 digipeater at Elk Neck, MD (the head of Chesapeake Bay). This remote system has been a bit ill for the past few weeks, but is now back on the air. Reports on local digipeaters were presented. W2FPY-7 is now operational in Hopewell, NJ, serving as a relay northward in NJ. A Camden, NJ location has been picked to serve the Philadelphia area better. In Harrisburg, PA, WA3KXG is now operational. In the Baltimore/Washington D.C. area, WB4JFI-5 and W3VD-5 are operational. W3GXT-5 is on experimentally in Baltimore, and K3JYD has plans for a southern MD digipeater that could link to Richmond, VA.

"The next topic of discussion was packet bulletin-board systems (PBBS). WB2MNF (NJ/Philadelphia area) and W3IWI (Baltimore/Washington area) are running the latest WØRLI MailBox software with automatic inter-PBBS message forwarding. In the Baltimore/Washington area, KS3Q provides backup to the W3IWI PBBS. WB4APR-5 serves as a gateway to 10 MHz, and occasionally W3IWI provides a teleport gateway through the AO-1Ø satellite. After sorting out PBBS user lists, there was considerable discussion of efficient use of current linking capability.

"It is becoming obvious that 145.01 MHz is getting clogged in the major metropolitan areas. We decided to recommend that users QSY to 145.03 MHz or 145.05 MHz whenever they have simplex paths available. It was also decided that the PBBSs should continue to operate on the 145.01 "primary" frequency until such time as the PRBS software supports QSY on request.

"The topic of beacon messages and CW IDs provoked a good discussion. Since the FCC has dropped the CW ID requirement for users, it was decided that we should recommend that all stations cease CW ID as soon as possible. There was some discussion (and a move to review the FCC rules) to see if dedicated digipeaters really have to have CW IDs -- after all, they identify themselves every time they repeat a packet. It was also agreed that all user beacons be discouraged except for PBBSs and Gateways that need to send "Mail for:..." beacon messages periodically. It was further agreed that there is no need for digipeaters to send beacon messages at all.

"A lot of hope was expressed that we would soon begin implementing 22Ø-MHz links for intercity communications. The long-haul links could profit from not having to contend with local-area users.

"All in all, it was a great way to spend a sunny Sunday afternoon. All attendees agreed that it

was well worth the 5Ø- to 1ØØ-mile round trip to get there."

From **W3IWI**

CONTINUING **TELEPORT** EXPERIMENTS

Experiments continued in November under the AMSAT/ARRL teleport STA.

ZL1AOX was linked to WB5EKU via an OSCAR-10 teleport station using the following links: ZL1AOX, Ian Ashley, Auckland, New Zealand, OSCAR 1Ø, Mode B; NK6K, Harold Price, Redondo Beach, CA, OSCAR 1Ø, Mode B; NK6K-2, Redondo Beach, CA, 145.36-MHz FM; WA6OZJ, Jim Johnson, Rolling Hills, CA, 145.36 MHz-FM; WB5EKU, Don Jacob, San Fernando Valley, CA, 145.36-MHz FM.

WH6AMX, Rick Dittmer, was linked (I'm resisting the urge to say he was teleported, which sounds too much like Scotty beamed him up.> via NK6K to the WB6YMH-2 remote CP/M system run by Skip Hansen on Palos Verdes, a local California hilltop. Several kbytes of data were moved under computer control from California to Honolulu.

Refer to Fig. 23, page 19-23 of the 1985 ARRL Handbook for the Radio Amateur for a diagram of a similar linkup using the KA6M teleport station. The teleport STA permits experimentation with fully automated teleports, and the coming months will see the development of teleports as "hands off" as a standard digipeater.

From **NK6K**

WESTNET NEWS

In more down-to-earth linking news, WESTNET has linked San Diego to San Francisco. For this winter, at least, WESTNET uses two audio hops to link Los Angeles to a point near Fresno. A key point in WESTNET is the WA6RWN station, a totally solar-powered mountain with interlinked "repeaters" on 5Ø, 144, 22Ø, and 44Ø MHz. This station is used as the northmost audio link for packet-radio communications. Initial testing succeeded in connecting northern and southern California digipeaters. This should allow access from the Mexican border to north of San Francisco. Extended testing, tuning, and aligning of the WA6RWN machine depleted its 4ØØ amp-hour batteries, so regular linking of North and South awaits a few more sunny days.

The audio paths will be replaced by digital links in the 22Ø- and 12ØØ-MHz bands in the spring. A more complete description of the current system will be forthcoming.

From NK6K

NTS PACKET NETS

We received the following from William Jochimsen, the Section Traffic Manager (STM) for the Southern Florida Section. Mr. Jochimsen is responding to Howard Goldstein, N2WX, who wanted to know how to gain formal NTS recognition of a packet traffic net.

"I perceive no difficulties in recognizing a packet-radio net, provided monthly net reports are received, as with any other traffic net. As for the message format for packet radio, I have recommended the preamble be on the first line, the address and phone number on the second line (separated by commas), the text on following lines, and the signature on a separate line at the bottom. This format was used recently in a local 'emergency' exercise and seemed to work fine."

So, if you want your message handling on packet radio to be recognized by NTS, simply send a monthly net report to your STM. It should be fairly easy to have your net report automatically generated and delivered.

Ed.

CIPBUS PACKET-USERS DIRECTORY

The Central Illinois Packet Radio User Society (CIPRUS) has completed the first edition of its Packet User's Directory, with 4ØØ stations listed. Greg Smith, N9AGC, tells us that the second edition cannot be far behind, since the CIPRUS data base now contains 5ØØ stations. The club was a bit surprised by the costs required to print and mail the directory, and asks that those who request the directory make at least a $5 donation. All funds go directly to CIPRUS to cover printing of the directory and to advance packet links in Illinois. The directory has three sections, an alphabetical listing of stations by call sign, a Zip code sorted listing of stations, and a description of each locality with frequencies, local-area nets, nets, clubs and other information.

Congratulations to CIPRUS !

Via **N9AGC**

LONG ISLAND PBBS

There is a new packet bulletin board system on 220.780 MHz, in Port Jefferson, NY, on Long Island. Norm Sternberg, W2JUP, is running a RTTY repeater on 223.18Ø/223.78Ø MHz. The PBBS operates simplex on the output frequency. The access command is "<cr><lf>W2JUP<cr><lf>" for a C 64 computer running the "Mailbox 64" program. This system will alternate some evenings with an Apple II "Super Ratt" system, with access code ":W2JUP<cr><lf>". The packet system uses an AEA PKT-1 TNC, and runs about 3Ø watts ERP at 330 feet above sea-level. [That's quite a mountain for Long Island. -- Ed.]

Via **HAMNET**

WØRLI MailBox SOFTWARE

As mentioned several places in this issue of Gateway, Hank Oredson, WØRLI has written a versatile PBBS called "MailBox." Far from being a run-of-the-mill PBBS, Hank's program has many innovative and useful features that make it outstanding.

The program runs on the Xerox 82Ø computer, with 2 disk drives and a TAPR TNC.

The bulletin-board section of **MailBox** stores messages, transmits broadcast packets telling who has messages waiting, and delivers your messages to you when you connect to the PBBS. The addition of automatic message forwarding, as mentioned earlier, makes this a powerful program.

The **MailBox** also has a “gateway” or “bridge” feature which allows the Xerox to control two **TNCs** on different **frequenc** ies. For instance, stat ions on 2 meters can use the gateway to connect to stations on an HF band like **1Ø** MHz. The gateway feature has been used to link stations from Arizona to Massachusetts.

To get a copy of **MailBox,** send an 8" disk, with return postage, to

Hank Oredson, **WØRLI**
19 N. Hill Rd.
Westford, MA 01886.

Ed.

REPRODUCTION OF GATEWAY MATERIAL

Material may be exerpted from Gateway without prior permission, provided that the original contributor is credited and Gateway is identified as the source.

Gateway: The ARRL Packet Radio Newsletter is published by the

American Radio Relay League
225 Main Street
Newington, CT **Ø6111**

Jeffrey W. Ward, **K8KA**
Editor

Larry **E.** Price, **W4RA**
President

David Sumner, **K1ZZ**
General Manager

Subscript ions are available to ARRL members and nonmembers, at the f0 1lowing rates:

For 25 issues of Gateway:
In the U.S.
ARRL Member $6.00
Nonmember $9 .OO
In U.S., Canada, and Mexico by First Class Mail
ARRL Member $11 .OO
Nonmember $14 .OO
Elsewhere by Airmail
ARRL Member **$14.00**
Nonmember $17.00.

Prices are subject to change without notice. Remittance must be in U.S. funds, and checks must be drawn on a bank in the U.S.

Gateway: The ARRL
Packet Rad io News letter
American Radio Relay League
225 Main St.
Newington, CT 06111

Nonprofit **Organizat** ion
U.S. Postage
PAID
Hartford, **Conn.**
Permit No. 2929

Gateway

Vol. 1, No. 10
Dec. 18, 1984

The ARRL Packet-Radio Newsletter

ATTENTION GLB DIGIPEATER USERS

If you are using a GLB PK-1 primarily as a digipeater, you should immediately get a software update from GLB. Software prior to Version 3.5 will not digipeat multiple frames with multiple digipeaters. If a GLB PK-1 is specified as part of a multirepeater path, only the first frame in any transmission will be repeated. This can exponentially increase the traffic on a channel where a PK-1 is a wide-coverage digipeater.

To get your software updated to Version 3.5, send a $10 deposit (or your current PROMs) to GLB. When you receive your new PROMs, send the old ones to GLB and you will get your deposit back.

This is an important problem, and should not be ignored by GLB users. The GLB address is

GLB Electronics
1952 Clinton St.
Buffalo, NY 14206.

From GLB.

MORE FROM GLB

In Gateway issue 8, we reported that GLB electronics, makers of the PK-1 TNC, were at work on the PK-2. Ed Jackson, an engineer with GLB, was able to confirm that preliminary engineering work is being done on the PK-2, but that it is primarily a "commercial TNC." More important for the radio amateur is GLB's development of the PK-R, an integrated VHF radio and TNC. On the RF end of the PK-R is a 25-watt crystal-controlled, highly stable FM rig. The packet equipment is a PK-1 TNC. For repeater operation, you simply connect the PK-R to a 12-V supply and an antenna, hit the reset button and you're on the air, For home-station use the rig has an RS-232C connector that goes to a terminal or computer. The PK-R is ready for type-acceptance tests, so look for it on the market before long.

Mr. Jackson also reports that a 200-Hz shift modification for the PK-1 is almost complete. This, along with an existing 300-bit/s modification will allow PK-1 users to join the growing number of HF packet users.

Ed.

TWO-METER BAND PLANNING

In several metropolitan areas, the amount of packet-radio traffic has exceeded the capacity of a single 1200-baud channel. On the east coast, the channel that is full is 145.01 MHz, and one of the most-asked questions is "what frequency do we go to next?" This question has no simple answer.

Ultimately, packet radio will move up to and above 220 MHz. When intercity "backbone" links are running on 220 MHz, the congestion on 2 meters will be eased. However, even when the network backbone is in place and experienced packet operators are moving to higher transmission speeds on higher frequencies, local traffic and "entry-level" packet operation will be on 2 meters.

So, the question remains: What frequency do we go to next? Some people on EASTNET have proposed a 20-kHz channel spacing, using 145.010, 145.030, 145.050, 145.070 and 145.090 MHz. Albert Hamilton, AG1F, Chairman of the New England Spectrum Management Committee points out that "Any rig produced in the last 6 years is perfectly capable of operating on a 15-kHz spacing without undue problems. While there are some real pieces of junk operating on 145.010 MHz now, that is no reason to give up sensible spacing." Using a 15-kHz spacing would provide six channels between 145.000 and 145.100 MHz -- one more channel than could be provided with 20-kHz spacing. We may need this extra channel in a few years.

If packet activity in your area is beginning to move away from 8 single 2-meter frequency, consider the arguments for a 15-kHz spacing. If you have any comments on this topic, address them to Gateway and

Albert W. Hamilton, AG1F
Chairman
New England Spectrum Management Committee
54 Hathaway Ave.
Beverly, MA 01915.

Ed.

RTTY-TO-PACKET GATEWAYS

Albert Hamilton, AG1F, who made the above suggestions concerning band planning, is also interested in working on a RTTY-to-packet gateway. Ideally, such a gateway would operate in real-time, allowing the RTTY user and the packet user to have a proper contact without knowing what mode the other operator was using. An alternative to this scheme would be to have a bulletin-board system that could be accessed by both packet and RTTY stat ions. Since there are already many stations equipped for Baudot RTTY, packet-to-RTTY gateways could bring interested amateurs in contact with packet radio while increasing the utility of packet networks for traffic handling.

If you have undertaken a packet-to-RTTY project, send the details to Mr. Hamilton and Gateway.

From **AG1F.**

PACKET IN PENNSYLVANIA

Gary Hoffmann, **AK3P,** sends us a summary of packet activity in central Pennsylvania. Gary got his TAPR TNC in June, and was the first station on the air in the Harrisburg, PA area. Shortly after that, Bob Warrington, **WB3FQL,** and Tim Shiogara, **WB3EYB,** were also in operation. Tim supplied a link to **WB4APR** near Balitmore, and the group gained access to **EASTNET.**

In October, Gary installed a wide-coverage digipeater above Harrisburg in the building of the Central Pennsylvania Repeater Association. The digipeater, running under the call **WA3KXG,** is in operation 24 hours per day, on **145.01** MHz. From its **1400-foot** elevation, **WA3KXG** provides a reliable link into Baltimore, Washington, and Virginia, via **WB4APR** and **W3VD.** A link into New Jersey has been established through **WB3FYL** and **W2FPY.**

There are presently about 5 stations on packet in the Harrisburg area, all using TAPR **TNCs.** Gary hopes that activity will increase in the near future, with links established into Scranton, PA and New York state.

Another item from Pennsylvania: Neil Sablatzky, **WA2WIM,** in Edinboro, PA (near Erie) is interested in becoming active on packet. His problem is that no one else in his area is interested. He would like to help link Northwest Pennsylvania with the eastern and southern parts of the state. He would also like to establish some contacts into the Buffalo, NY area. Neil would like to hear from anyone near Erie who is working on packet networking. He can be reached on the **WB3USH** repeater on 146.76 MHz, where he will be listening on packet when his TAPR TNC arrives. You can write him at:

201 Granada Drive
Edinboro, PA 16412.

From **KC2WZ** and **AK3P**

MIDWEST NETWORKING

The packet-radio network that has grown up around central Iowa has finally reached the **Iowa-**Minnesota border. Lyle Monson, **WØFQ,** is active from Spirit Lake, IA.

Lyle wants to know what frequencies are being used in southern Minnesota, and is looking for some active stations to hold schedules with. He will be able to connect to the rest of the Iowa packet network via **WØZVY,** the wide-coverage digipeater in Guthr ie Center, IA.

The Central Iowa Technical Society **(CITS)** is planning a **220-MHz** intercity network. They are looking for a station around Mason City, IA to extend this backbone to the north.

From **WØRPK.**

MT. BEACON DIGIPEATER

The gaps in **EASTNET** are beginning to be filled in. The addition of the **WB2KMY** digipeater on Mt. Beacon, NY will help to tie both Boston, MA and Newington, CT into the southern portion of the network.

The Mt. Beacon digipeater, on 145.01 MHz, is the first part of an ambitious project being undertaken jointly by the Mt. Beacon Amateur Radio Club and the Putnam Emergency Amateur Repeater League **(PEARL).** When the project is complete, there will be two complementary digipeaters in the mountains around Poughkeepsie, NY. One of the machines will be on Mt. Beacon, a site that has good coverage to the west. The other will be on Mt. Ninham, with a clear path to the east. Tht two clubs will be working together to **minimize** interference between the two digipeaters maximize the coverage provided by the pair.

The call used at the Mt. Beacon site is **WB2KMY-1,** and the call at the Mt. Ninham site will be **KG1O-**9. If you are located in northern New Jersey, Northeastern Pennsylvania or Western New England, check out these new facilities. (Leave reports of coverage on the **WA2RKN-2 MailBox.)**

If you are not in New England, there is still something of value here; packet repeaters, unlike Z-meter voice repeaters, must coexist on a single frequency. If several clubs in one area are interested in packet radio, it is important that they work as a team to provide valuable and useful coverage. Lack of cooperation can only lead to a congested local-area network.

via **WB2KMY.**

WESTNET IN MONTEREY

Packet operations in Monterey, CA, have improved due to the addition of a **6-element** Yagi at the **K6LY** digipeater site. Buzz Shaw, **WA1NHP,** reports that the Monterey packet operators are also looking into a site on Mt. Toro, at an elevation of 3500 feet. This would be a real improvement over the current site, which is only 110 feet above sea level. However, even at the low elevation, stations using **K6LY** can usually connect to Hank Magnuski, **KA6M,** and use his Data General computer system.

Via WA1NHP

ILLINOIS PACKET MEETING

Twenty-five people, representing networks in Central Illinois, Southern Illinois, Chicago and St. Louis, attended the December 2 meeting of the Central Illinois Packet Radio Users Society **(CIPRUS).** The administrative portion of the meeting resulted in a statement of the club's intent "to promote packet radio throughout Illinois by acting as a clearing house for packet inf ormat ion, and to work for reliable packet communication within Illinois."

After electing club officers, those present took up a discussion of packet-radio operating practices. Since packet radio is new to amateur

what constitutes "good operating procedure." In particular, there is still no agreement as to when it is appropriate to send out CQ or QST messages. Broadcasting such frames through wide-coverage digipeaters has started to significantly slow traffic on several crowded networks. However, those at the CIPRUS meeting agreed that QST packets used as propagation indicators or to carry news bulletins were reasonable. They also decided that CW identification is not necessary and should be discouraged. [Close examination of sect ion 97.84 of the FCC Amateur Radio Service Rules shows this to be an acceptable decision.-- Ed.] It was observed that application or common sense by packet operators would significantly reduce network congest ion.

The CIPRUS meeting closed with the distribution of the the national Packet User's Directory as compiled by N9AGC and available through CIPRUS.

Via K9NG.

PACKET MAP

Paul Barnett, NØCRN, placed the following message on DRNET:

"I'm compiling information on existing amateur packet networks to use at an upcoming packet-radio demonstration. I want to be able to show a large map of North America that is sufficiently detailed to explain multiple digipeaters and show how packet-radio networking is progressing.

"I would like everyone to send me information about their local network, so that I can generate the map. Please confine the listed stations to those with wide coverage. I'm not particularly interested in the 50 stations within 25 miles of each other in a metropolitan area. What I am interested in is the stations that make up a link into another metropolitan area.

"I'd like to hear about all forms of networking: VHF (EASTNET, SOUTHNET, WESTNET), HF (particularly gateways and scheduled traffic handling operations) and OSCAR teleports.

"Confine your descriptions to reliable links, noting those that are not available 24 hours per day. Please limit information about planned links to those that will be a available by the end of the year.

"When I get the map done, I will try to reduce it to a single page and make it generally available.

Send information about your local network to
Paul Barnett, NØCRN
130 Demont Ave. East, Apt. 255
Little Canada, MN 55117

or to Gateway at the ARRL.

Via DRNET.

PACKET APPLICATIONS: WEATHER

K6AAG>QST:HIDESERT WX 1100/52/SCTRD/VIS10HZ/SSE10-15/29.71/"Robbie"

Although running in an attended and manual mode now, the above is an example of vhat could become an automated service as packet radio continues to grow and rules continue to change.

For those who need help in interpretation (as I did), the above QST frame says that the veather in Southern California's high desert region at 11:00 was 52 degrees, scattered clouds, visibility 10 miles with haze, winds from the south southeast at 10-15 knots, barometer 29.71 inches. Bobbie Wohosky, K6AAG, the originator of the message, is active in the California veather net on 75 meters. He currently summarizes the information gathered by this net and distributes it via BTTY. He is now looking into ways to use packet radio as well. It is probably a safe bet that amateur-gathered weather information will be part of the data stream as regional and national digital netvorks take shape.

[If you have thought of any interesting applications of packet radio, please send them along to Gateway. -- Ed.]

From NK6K.

PACKET READING

Paul Newland, AD7I, points us toward IEEE's Communications Magazine, an IEEE monthly publication that features tutorial technical discussions about communications concepts and systems. (Don't confuse it with IEEE Transact ions on Communications, a mathematical and theoretical journal.) The December issue of Communications Magazine contains the following articles of interest to packet-radio enthusiasts:

"Automatic-Repeat-Request Error-Control Schemes," by Shu Lin, Daniel Costello, Jr., and Michael Miller. Error detection incorporated with automatic-repeat-request (ARQ) is widely used for error control in data communications systems. [Including packet radio. -- Ed.] This method of error control is simple and provides high system reliability. If a properly chosen code is used for error detection, virtually error-free data transmission can be attained. This paper surveys various types of ARQ and hybrid ARQ schemes, and error detection using linear block codes.

"Improving on Bit Error Rate," by Virgil I. Johannes. On the semantics of the term -- its invalidity in the absence of a specified averaging period.

"Digital Signaling Techniques," by William Stallings. A comparison of various encoding techniques with an eye to superior performance.

Also look into the November, 1984 issue of Proceedings of the IEEE. This special issue on satellite communication networks contains an article titled "Packet Switching for Mobile Earth Stations via Low Orbit Satellite Network." Remember, PACSAT is a low-orbit packet satellite.

R.F. Design magazine for November/December, 1984 contains the conclusion of a series called "RF Modems." This article addresses the design and use of frequency agile modems. Several block diagrams and schematics are presented.

The December, 1984 issue of Telecommunications is a special issue on data communications systems, and contains several articles that packet experimenters might find interesting.

Via AD7I, KA9Q, Ed.

MISSING GATEWAY

When you subscribed to Gateway, you probably subscribed for 25 issues. A few quick calculations will tell you that there should be 26 issues in a 52-month year. The missing issue is to allow me to take some time off around the New Year. Therefore, there will be no Gateway dated January 1, 1985. Look for Gateway issue 11 on January 15. I hope that you all have safe and enjoyable holidays.

Jeff Ward, K8KA

REPRODUCTION OF GATEWAY MATERIAL

Gateway: The ARRL Packet Radio Newsletter is published by the

American Radio Relay League
225 Main Street
Newington, CT 06111

Jeffrey W. Ward, K8KA
Editor

Larry E. Price, W4RA
President

David Sumner, K1ZZ
General Manager

Subscriptions are available to ARRL members and nonmembers, at the following rates:

For 25 issues of Gateway:
- In the U.S.
  - ARRL Member $6.00
  - Nonmember $9.00
- In U.S., Canada, and Mexico by First Class Mail
  - ARRL Member $11.00
  - Nonmember $14.00
- Elsewhere by Airmail
  - ARRL Member $14.00
  - Nonmember $17.00.

Prices are subject to change without notice. Remittance must be in U.S. funds, and checks must be drawn on a bank in the U.S.

Gateway: The ARRL
Packet Radio Newsletter
American Radio Relay League
225 Main St.
Newington, CT 06111

Nonprofit Organization
U.S. Postage
PAID
Hartford, Conn.
Permit No. 2929

Gateway

Vol. 1, No. 11
Jan. 15, 1985

The ARRL Packet-Radio **Newsletter**

PACKET NETWORK COORDINATION

The frequency-coordinating body for Northern California, NARC, has officially coordinated the following frequencies for packet radio: 145.01, 145.43, 145.05, 145.07, 145.69, 223.56, 223.50, 223.6, and 441.000 MHz. It is unclear what this means for current activity on 146.58 MHz.

NARC has also requested that all "mountain-top digipeaters placed on the assigned frequencies be coordinated through NARC, to insure proper record keeping." This brings up the question of where the jurisdiction of the frequency-coordinating council ends. Hank Nagnuski, KA6M, of the Pacific Packet Radio Society makes the following suggestions:

"The [frequency-coordinating councils] are dealing with very real problems of crowded frequencies, overlapping repeater coverage, intermod, etc. It is a good idea that packet frequencies be coordinated through such councils.

"However, once frequencies have been allocated to packet radio, the jurisdiction of the frequency-coordination council ends, and another type of advisor is needed. This advisor must help track and promote the rational growth of packet radio equipment, repeaters, gateways, and protocols on the assigned frequencies. The advisor may eventually supervise the growth of the packet network.

"Therefore, I would like to propose a new term for amateur radio: 'Network Coordinating Agent (NCA). The NCA for a region deals with the problems mentioned above, and also represents its area in national discussions of protocols and other issues.

"The Pacific Packet Radio Society, because it is the oldest and largest packet-radio club in the Bay area, has become the Network Coordinating Agent for the San Francisco area."

The need for NCAs has also been recognized on the East Coast. Groups from New England and from the Mid Atlantic states agree that congestion on 145.01 MHz must be dealt with in an organized manner. The first EASTNET NCA is the newly-formed Mid-Atlantic Packet Radio Council. This council will coordinate for MD, DE, Southeastern PA, Southern NJ, Northern VA, and Washington D.C.

Send Gateway any comments that you have on the need for NCAs.

From KA6M.

HIGH-SPEED PACKET PROGRESS

At the January 12 meeting of the Chicago Area Packet Radio Association, Steve Goode, K9NG, demonstrated his 9600-bit/s, 220-MHz modems. When the plans for Steve's modems become available (perhaps early this spring), it should be possible to reduce some of the congestion on 2-meter metropolitan packet networks.

The modems are built around inexpensive Hamtronics FM-5 transmitter and receiver strips. The addition of a "data filter" in the modulator gives the FSK signal desirable spectral characteristics, while a corresponding circuit in the demodulator converts the data back to something that a TNC can understand.

Congratulations to Steve for spending the great amount of time necessary to design and debug the modems. The entire packet-radio community will benefit from his work.

Ed.

ACTIVITY ON THE UoSAT DCE

There has been a lot of activity with the Data Communications Experiment (DCE) aboard the UO-11 satellite this week. In preparation for a possible demonstration of the PACSAT store-and-forward "mailbox" concept, software for a prototype message handling system has been uploaded to the satellite. Monitor UO-11 bulletins to look in on the action.

Via NK6K.

HDLC ADD-ON AVAILABLE

Those of you experimenting with level-3 packet networking should note that a High-level Data Link Controller (HDLC) add-on is now available for the Xerox 820 computer. This piggyback "FAD board" performs many of the complicated tasks involved in transmitting and receiving AX.25 frames. This development is particularly important because the ARRL Ad Hoc Committee on Amateur Radio Digital Communication agreed that the Xerox 820 with the HDLC add-on would be the standard computer system for level-3 software development and testing.

TAPR now has a limited number of the add-on boards available, intended primarily for those doing level-3 experiments. Each pc board comes with a parts list, a schematic, and a brief survival guide. All told, there are only 2 pages of documentation; this unsupported product is offered

only as a service. Parts for the board are not available from TAPR at this time. If you are still interested, the price is $25.

Via **WA7GXD.**

22Ø-MHZTERS

Scott Miles, WB6PQM, is establishing two new 22Ø-MHz digipeaters in the San Francisco Bay area. The new simplex machines will be running on 223.58 MHz, and are intended to serve the needs of RACES in Contra Costa county. The digipeaters will be based on GLB PK-1 TNCs.

Rich Collins, NT6V, reports that he and David Bly, WD6EVM, have installed a digipeater on 220.58 MHz in San Leandro, CA. This site, made available by the Alameda County RACES group, provides good coverage of the Bay area.

Via **KA6M.**

VADCG AX.25 **DIGIPEATERS**

Those of you who are using VADCG TNCs as digipeaters should be interested to know that Les McClure, W3GXT, and Mike Bruski, AJ9X, have fixed many of the program bugs that make these digipeaters fail regularly. In addition, they have installed some software and hardware timers' which make certain that the program can recover from any faults that occur.

Mike and Les would like to hear what other groups around the country are doing with the venerable old VADCG boards. If you want a copy of their "bullet proof" VADCG AX.25 digipeater code, send an 8" SSDD disk to

Tom Clark, W3IWI
6388 Guilford Rd.
Clarksville, MD 21029.

Via **DRNET.**

MAIL FORWARDING **IN EASTNET**

Several more stations have joined those capable of automatically forwarding and receiving messages via packet radio. In the Mid-Atlantic section of EASTNET, Gary Hoffman, AK3P, is operating WØRLI's MailBox program. From Gary's station in Hummelstown, PA, traffic is forwarded through 3 digipeaters to the W3IWI MailBox that serves Baltimore and Washington. The W3IWI MailBox is able to forward messages to WB2MNF in southern New Jersey. At a recent meeting, members of the Mid-Atlantic Packet Radio, Council (MAPRC) agreed that "the linked PBBSs are providing a very necessary communications conduit for MAPRC."

In the northern portion of EASTNET, Jerry Koniecki, WA2RKN, located in Hyde Park, NY, is providing MailBox service to stations from Long Island up the Hudson River valley. The W1AW packet bulletin-board system has been made fully compatible with the WØRLI forwarding software and is in communication with both WA2RKN and WØRLI (in Boston).

A long-standing goal of EASTNET is to link New England with the Mid-Atlantic. It looks like a well-placed station in Northern New Jersey would be able to complete that connection.

For further information on the message forwarding protocol being used, contact Gateway or Hank Oredson, WØRLI.

Via W3IWI, Ed.

CENTRAL STATES PACKET NET

The Central States Packet Radio Users Group Net, sponsored by the Central Iowa Technical Society (CITS) has moved from 4Ø meters to 75 meters. The new frequency for the net is 3865 kHz, and the time remains 15ØØ CST on Sundays. The net will continue to discuss items of general interest to packet-radio operators, but will concentrate on discussion of linking and networking in the Midwest.

The first session of the net on its new frequency was January 13, and Ralph Wallio, WØRPK, reports that stations checked in from Michigan, Illinois, Iowa, and Nebraska. Experimenters from Nebraska would like to encourage stations from the Rocky Mountain area to check in, and stations in Illinois would like to hear from packet operators in Indiana.

Ralph suggests that Midwestern packet enthusiasts adopt 3865 kHz as an informal "calling frequency." Operators who want more information on packet radio, or who are having trouble with their packet equipment could just give a call on 3865 kHz and get their questions answered.

From WØRPK.

NTS NETS AND PACKET RADIO

Eastern Massachusets Section Manager Luck Hurder, WA4STO, adds these facts to the ongoing discussion of NTS recognition for packet-radio traffic nets:

"I was startled to see the article regarding NTS packet nets in the December 4 issue of Gateway.

"According to the article, the STM of the Southern Florida section has stated that monthly net reports are the key criteria for acceptance of a packet net as a NTS net.

"I certainly don't wish to be a wet blanket about all this; please understand that I am an active packet operator. To be fair to anyone wishing to attempt to have their packet net become an official part of the National Traffic System, however, there are a few ground rules that must be taken into account. Some of them are:

"1. The Net Manager must be appointed by the Section Manager or his Section Traffic Manager.

"2. The Section net must adhere to section boundaries.

"3. The net must function on the basis of a prescribed, orderly traffic flow.

"4. The net must interface in a prescribed manner

8) Expansion interface for Board Three (described below).

"This will run on 5 v, DC and has RS-232 compatible port6 for the two asynchronous channel6 that are part of the 64180 microprocessor. The serial interface will meet the proposed **WESTNET** es tandard .

"But, you may ask, what good is a NNC without modems? Glad you asked that!

"Board Two, to go to St. Louis next week, consists of:

1) Multiple **XR2206/XR2211** modems. [Like the modems in existing TAPR **TNCs.]** Each modem will have a clock generator, a state machine and a tuning indicator. Board size constraints will determine whether we only get two modem6 or if we can squeeze on four modems per card.

"Yes, these are only **1200-baud** (or **300-baud)** modems . But, the local users need a port or two to get in **(1200** baud) and long-haul stuff is going to be HF for a while to come (300 baud now, perhaps **1200** later).

"Both of these boards to be sized per the **WESTLINK** standard, so that they can screw on the side of a **5.25-inch** floppy-disk drive.

"Why a floppy drive? I'm glad you asked that, too !

"Board Three is a plug-in **floppy-disk** interface! The I/O is mapped to be compatible with the **SB180** to allow a simple port of the **"Z"** system [disk operating system]. Thus, the NNC can become its own software development engine, and the hard work of placing a decent operating system on the NNC is already done and readily available at a reasonable price .

"If we are lucky, all boards will be laid out by the end of September. Prototype boards should be populated in October, then debugged by hardware types while the software types **(I** hope) will get cranking on some level-three [networking] software.

"Thank you each and every one for your inputs to date. Keep the comments coming. **Happy packeting!"**

[If you are interested in what the 64180 microprocessor is like, be sure to read the referenced Byte article. For further, still introductory, discussion of some of the computer hardware concepts introduced in the above item, read chapters 8 and 19 in the 1985 ARRL Handbook. -- Ed.]

Via DRNET.

MAPRC MEETING REPORT

The Mid-Atlantic Packet Radio Council (MAPRC) met on September 11, in conjunction with the Gaithersburg, Maryland, hamfest. The following item is condensed from a report filed by Mike Chepponis, **K3MC,** and Bob Hoff man, **N3CVL.**

About 40 hams were present. The first hour of discussion concerned MAPRC organization. A copy of the proposed MAPRC constitution was distributed and **discussed.** It was resolved to collect $24 per year membership dues with $50 per year dues for clubs. Clubs would get two votes, and individual member6 would get one vote. The official purpose of MAPRC is to promote linking in the Mid-Atlantic area and to make funds and expertise available to install and maintain digipeaters. Projects funded by MAPRC will include digipeaters in areas that are critical to network growth, but without large populations of **packeteers.**

The group then discussed **EASTNET** and **MidNet** coverage. Bob Hoffman, **N3CVL,** distributed copies of a map of **MidNet** and three **possible** paths between **MidNet** and **Eastnet** were analyzed:

A) Through Northern Pennsylvania and the Buffalo area. This link would enter **EASTNET** territory at **K3RLI** in Wilkes Barre, Pennsylvania.

B) Through Harrisburg and into Philadelphia. Gary Hoffmann, **AK3P,** is already working on this path.

C) Through West Virginia. This route would enter **EASTNET** at Cumberland, Maryland, via **N8FJB,** near Martinsburg, West Virginia.

The discussion turned to multiport digipeaters and whether it is best to use the **440-MHz** band or the **220-MHz** band for **9600-bit/s** links. **440** MHz is preferred by some, especially Brian Lloyd, **WB6RQN,** because high quality commercial gear is available for that band. Groups without well-equipped RF labs have had problems with the Hamtronics FM-5 transceivers used with the **K9NG 9600-baud** modems. On the other hand, many folk6 already have some kind of **220-MHz** equipment, and Bob, **N3CVL,** pointed out that there are other **220-MHz** transceiver boards that are fairly easy to align. It was also noted that **WB4JFI-5** is already on **220** MHz. There was some confusion as to what frequencies are allocated on 220 MHz and 440 MHz for packet. Some people are reluctant to use the high-speed **(100-kHz)** channel6 allocated on 220 MHz for narrow-band **(20-kHz)** modems. Others, pointed out the advantage of being able to change to higher speeds in the future without the need to change channels. It was resolved to use 220 MHz for the backbone.

Will Xerox 820s be used as the backbone boards? [In the near future, the answer is probably "yes," because the multiport digipeater code runs on the Xerox 820. -- Ed.] Discussion **was** inconclusive, but it was noted that the 820 [since it uses an **8-**bit microprocessor] is running out of steam. Possible alternatives are machines based on the 8088, like the IBM PC, or machines with large linear address spaces, like the 68000. When choosing a second-generation networking controller, we must pay close attention to the availability of inexpensive development tools -- in particular, compatible computers on which software can be written and tested.

Brian Lloyd and Phil Karn, **KA9Q,** discussed some technical aspects of the network. For example, making FRACK [the time that a TNC waits before retransmitting a packet] and DWAIT [the time that a TNC waits after the channel is clear before transmitting] fairly large number6 really help6 out on busy or noisy channels. There was some

Gateway

Sept. 17, 1985
Vol. 2, No. 3

The ARRL Packet-Radio Newsletter

LOW-POWER TNC FROM GLB

GLB Electronic6 has released its PK-1L TNC. The PK-1L is an enhanced, CMOS version of the successful PK-1 TNC. Running with a CMOS Z80A microprocessor, the PK-1L draw6 only around 25 ma, and it can even be run on a 9-v transistor battery. GLB engineer Ed Jackson, WB2OIF, estimates that the PK-1L would run for about 20 hour6 from a 9-v alkaline battery. This TNC should be great for solar-powered operation. As well as supporting all the features of the PK-1, the PK-1L will have an on-board watchdog timer, battery-backed-up RAM to store operating parameters in case of power failure, standard DB-25 connectors for radio and terminal I/O, a bit on the radio connector that indicate6 whether the TNC is connected, and two spare I/O bits available for later expansion. The TNC comes assembled and tested, in a 4.5 X 6 X l-inch, all-metal cabinet. It should make a great portable or remote station.

Prom WB2OIF.

TAPR NETWORK CONTROLLER

The availability of several commercial TNCs has spurred the growth of packet radio. In most areas, several new calls are heard on the air each week, and Gateway is getting between 50 and 100 new subscribers per issue. How will these new users be served by the existing network? Moreover, how will network continue to offer more services in the face of a steadily growing user community? The single-frequency amateur packet-radio network is already nearly overloaded in many metropolitan areas. When the network become6 overloaded, the packet community will look to its experimenters for new hardware and software to keep packet radio growing.

Several groups of experimenters, both formal and informal, are investigating the various hardware and software choice6 that face the amateur packet-radio community. The necessary software include6 protocols for network (also called ISO layer 3) and transport (ISO layer 4) services, and standards for addressing, routing and mail forwarding. Hardware will be needed for sophisticated mountain-top digipeaters, remotely operated PBBSs, satellite teleports and HF-to-VHF gateways, While Gateway is a newsletter and not a technical journal, much of the packet-radio news in the next few months will concern the projects and standards mentioned above. When these topics are covered in Gateway, we will try to define terms as we introduce them, to stick to newsworthy items, and to list sources of more extensive information.

One of the groups interested in what is broadly called "networking hardware," is TAPR. Now that several hundred TAPR TNC 26 have been shipped, TAPR is turning to the design and debugging of a Network Node Controller (NNC). TAPR president Lyle Johnson provide6 the following overview of the TAPR NNC project.

"To keep everyone in the loop, here is the present status of the TAPR NNC hardware project.

"'The schematics are in St. Louis at Interconnections, the company that does all the CAD (computer-aided design) layout work for TAPR. If all goes well, we should have artwork for all three boards by the end of the month!

"Board 1 is the NNC itself. Its present configuration is a6 follows:

1) HD64180 microprocessor. This is the CMOS Z80 superset chip with on-chip DMA (direct memory access), dual UARTs [for asynchronous communication6], 16-bit timers, MMU (memory management unit) [to manage 512 kbytes of memory] and a clock. This is the same microprocessor that is featured in "Build the SB180 Single-Board Computer" in the September issue of Byte magazine.

2) Dual SIO/2s. This allows four channels of HDLC (high-level data link control) capability. [HDLC is necessary for packet operation.] One SIO may be configured (via push-on jumpers) to have either or both of the channels operate DMA.

3) One PIO (parallel I/O) chip. This provide6 a parallel printer port and several lines to fiddle with (for bells, whistles, buzzer6 and item 4).

4) A battery-backed-up real-time clock.

5) An SCSI interface, which allows this board to communicate at high speed with other nearby devices. This will allow the NNC to be a smart Level Two "front end" for a later board that can handle all the networking and transport functions when the network outgrows the capacity of the 64180.

6) Eight byte-wide sockets, for 64 kbytes of battery backed-up RAM (bbRAM) with jumper selection for 256k bytes of bbRAM!

7) Eight more byte-wide sockets mapped for 32-kbyte parts. ..This allows the full 1/2 Mbyte of memory to be put on the board.

packet radio. As this activity pick6 up, don't forget to send us item6 of **interest. Gateway** would like to supply up-to-date packet-radio news. To do that, we need to know what is going on in your part of the network. You can send this information through the mail, via DRNET or via Compuserve **HAMNET** (ID 75105,737). If you have any comments on how we can better serve you through Gateway, send those along too.

This copy of Gateway was delayed by the Labor Day holiday. Previous issues of Gateway, however, should have been delivered regularly -- none more than a day or so late. If you are having erratic delivery, especially if your Gateway comes via First **Class** Mail, please send **us** a letter detailing the problem. If your Gateway is mailed at bulk rate, there is no way to combat delivery de lays.

Ed.

REPRODUCTION OF GATEWAY MATERIAL

Gateway: The **ARRL** Packet Radio Newsletter is published by the

American Radio Relay League
225 Main Street
Newington, CT 06111

Jeffrey W. Ward, **K8KA**
Editor

Larry E. Price, **W4RA**
President

David Sumner, **K1ZZ**
Executive Vice President

Subscriptions are available to ARXL member6 and nonmember **s,** at the following rates:

For 25 issues of Gateway:
- In the U.S.
  - ARRL Member $6.00
  - Nonmember $9.00 ,
- In U.S., Canada, and **Mexico** by **First** Class Mail
  - ARRL Member $11.00
  - Nonmember $14.00
- Elsewhere by Airmail
  - ARRL Member $14.00
  - Nonmember $17 **.00.**

Prices are subject to change without notice. Remittance must be in U.S. funds, and checks must be drawn on a bank in the U.S.

Gateway: The **ARRL**
Packet Radio Newsletter
American Radio Relay League
225 Main **St.**
Newington, **CT 06111**

Nonprofit Organization
U.S. Postage
PAID
Hartford, Conn.
Permit No. 2929

advantages, the digipeater and the TNC. The second part of the article will describe packet-radio equipment and applications.

John's article includes a **sidebar** on the his tory of packet radio, which shows how radio amateurs have influenced and developed packet-radio t**echno** logy. At the end of the **sidebar,** John says that he "believes those of us in the business radio environment would do well to avoid encroachment on [the **220-MHz]** and other amateur experimental bands. The results of work carried out on amateur frequencies may yield decided benefits for the commercial sector." The editorial comment in the magazine that carried John's article states: "We welcome the opportunity to tell the story when amateurs help commercial users by way of experimentation. We avoid contrasting amateurs with 'professionals."'

Packet radio is bringing Amateur Radio a lot of favorable attention from communications professionals. Congratulations and thanks to John Gates and the others who have had papers about amateur packet radio published recently in the professional press.

Ed.

CONNECTICUT PACKET CLUB FORMED

On September 3, the Southern New England Association of **Packeteers** (SNAP) adopted a constitution and formally became a club. SNAP hope8 to organize the growth of packet radio in Connecticut. The club will meet on the first Tuesday of every month at **7:30** PM at ARRL Headquarters, Newington, Connecticut.

For further information, write:

Jon Bloom, **KE3Z**
c/o ARRL
225 Main Street
Newington, CT **06111**

From WA2FTC.

JAPANESE PACKET

The following update on Japanese packet radio comes from **Tac** Okamoto, **N6MBM/JA2PKI. Tac** is temporarily living in Irvine, California, **and** is a regular on Compuserve **HAMNET.**

"You are interested in packet activity in Japan. Hummmm... Where should I start? A couple of years ago, a group called **JAMSAT** received a pair of TAPR beta-test TNC boards. As you may know, **JAMSAT** is a group like **AMSAT,** working to build the first Japanese amateur-radio satellite, JAS-1. JAS-1 has a packet-radio transponder [See Gateway issue **23].** When TAPR started distributing the TNC 1, **JAMSAT** bought **6** of them to use as development tools for JAS-1. Those **TNCs** were also used for **JAMSAT's** information network, JASNET. JASNET was the first packet-radio net in Japan. Now **JAMSAT** has finished hardware for two JAS-1 satellites. Yes, **JAMSAT** has built two **JAS-1s.** One of them will be launched in August, 1986. [The other one will be a back up. -- Ed.] So, the packet activity in Japan was started by **JAMSAT** with help of **AMSAT** and **TAPR.**

"Can Japanese amateurs buy a Japanese TNC? Yes, they can. **A** company in Japan bought a manufacturing license from AEA and is making a TAPR TNC **1/AEA** PKT 1 clone. The TNC costs around **$300.**

"How many **TNCs** are in Japan? I don't know exactly, but there may be 300 or more. Do they send and receive Kanji [Japanese alphabet] via packet? Yes, there is even a a Kanji **BBS!"**

Via HAMNET.

K9NG MODEM PROJECT

The modem design presented by Steve Goode, **K9NG,** at the Fourth Amateur Radio Computer Networking Conference generated a lot of excitement among packet-radio experimenters. The **9600-bit/s** modem is necessary for planned intercity "trunk lines" or "backbones." Since Steve's presentation at the conference, TAPR has been working on making available a PC board and a kit of hard-to-find parts for the circuit. The project took a little longer than expected, but the PC boards have been tested, and the final package should be ready for shipping before the end of this month. The PC board is designed to connect a TNC and a Hamtronics FM-5 **220-MHz** transceiver. Other transceivers can be substituted, with appropriate **modif ications** to the modem board. TAPR will not be selling a complete kit of parts for this circuit. The boards will probably come with the critical capacitors and the state-machine PROM. This will not be a step-by-step project for the beginning kitbuilder, but the documentation provided with the board and the article that Steve submitted to the Proceedings of the Fourth ARRL Amateur Radio Computer Networking Conference **(available for** $10 from the ARRL) should allow experienced builders to assemble and test the modem, When the kits are available, announcement will be made in Gateway and on packet-radio and telephone bulletin boards.

Ed.

HAWAIIAN ERRORS

In the last issue of Gateway, we incorrectly stated that Maui is the largest of the Hawaiian islands. Several people have pointed out that Hawaii is the "Big Island" and Maui is the **second-**largest Hawaiian island. The link discussed in the Gateway item was between **Hilo,** Hawaii, and Kalaheo, Kauai. Maui was not involved.

Ed.

GATEWAY ADMINISTRATIVE NEWS

Now into its second year, Gateway is as successful as we had hoped it would be. There are now over 860 paid subscribers to the newsletter, and several hundred officials of NTS and other ARRL field organizations receiving complementary copies. We believe that another thousand people receive electronic copies of Gateway from packet-radio or telephone bulletin boards.

Now that the summer is over**,** people will be retu rning to their shacks and getting back into

Bill Stash, WA3AOQ
421 Daily Drive
North Huntington,, PA 15642.

From AK3P.

PACKET SOFTWARE AVAILABLE ON BBS

The WORLI MailBox/Gateway program and the KE3Z multiport digipeater program are now available for downloading from the Ham Radio Net BBS in Newington, Connecticut. Ham Radio Net, run by Ed Raso, WA2FTC, can be accessed at either 1200 bauds or 300 bauds, and it is usually in service around the clock.

All cf the files necessary for configuring and running version 10.0 WORLI MailBox and Gateway are on the Ed's system, as are all of the files that you would receive if you sent a disk to the ARRL for the multiport digipeater code. These files are stored in File Area 2 of the BBS -- the Packet Radio Section. Because the programs and documentation are large, they have been squeezed and placed in library files. The program that will unsqueeze and separate them is the CP/M utility NULU11. NULU11 is also in File Area 2 on the Ham Radio Net.

Distributing WORLI's software electronically should allow new versions of the popular MailBox and Gateway to spread rapidly. Hank Oredson, WORLI, has put a lot of time into support of his software, and the latest release, Version 10.0, has several useful new features.

You can call Ham Radio Net at:

203-665-1114.

Via WA2FTC, KE3Z.

EASTNET UNIX UPDATE

Several improvements have been made to the UNIX-based news and mail gateway run by Jim Kutsch, KY2D. [See Gateway Issue 19.] Jim reports that the system, running as KY2D-2, is now operating 24 hours per day on 145.01 MHz. It can be reached reliably through the WA2VKH, WA2SNA-2 and WB2VTN-1 digipeaters. There are 10 frequent users of KY2D-2 as well as several casual users.

Jim has recently developed and installed software that will accept automatic forwarding of mail and bulletins from systems running the WORLI MailBox program. Mail received by KY2D-2 can be further forwarded into the USENET (an international network of UNIX systems), if the correct USENET address is included as the first line of the message.

Right now, mail is accepted only from the WA2VKH MailBox, but the system can be instructed to receive mail from any MailBox. You do not need to modify the WORLI software in order to forward to KY2D-2.

[UNIX is a trademark of AT&T Bell Laboratories.]

From DRNET.

TUNING INDICATOR FOR HF PACKET

If you are frustrated by the difficulty of tuning in HF packet stations, you may be looking for a good FSK tuning indicator. An interesting crystal-controlled digital tuning indicator is described in an article called "RTTY Tuning: The New Solution," by John Langner, WB2OSZ in the March 1983 issue of 73 magazine. John also discusses the design in.he August issue of the NEPRA PacketEar, newsletter of the New England Packet Radio Association. The tuning indicator is actually a simple frequency counter that display6 the mark and space frequencies on an LED bar graph. It look6 useful, and it should be inexpensive. If you read the article and want to build one of the indicators, Jon has a few PC boards available for $10.

Contact:

John Langner, WB2OSZ
115 Stedman Ct.
Chelmsford, MA 01824.

From The NEPRA PacketEar.

GEORGIA CLUB ADDRESS

The address for the Georgia Radio Amateur Packet Enthusiast Society (GRAPES) that was previously published in Gateway is no longer valid. The new address is:

GRAPES
P.O. Box 1354
Conyers, GA 30207.

Ed.

HF GATEWAY LIST

Don Simon, NI6A, is compiling a list of HF gateway stations. If you operate an HF gateway, even part time, send a letter to Don. Be sure to tell him your station callsign, location, coverage and hours of operation.

Don is also discussing with William Smith, W7GHT, the use of packet radio to pass messages to Australia and New Zealand. The two are looking for HF packet stations that might be able to connect to stations in those countries.

If you can assist with either of these projects, contact:

Don Simon, NI6A
2327 Alva Avenue
El Cerrito, CA 94530.

From NI6A.

MOBILE RADIO TECHNOLOGY ARTICLE

If you are looking for an article that describes packet radio to people who are familiar with communications but not with amateur radio, look at "Packet Radio Combines Computer, RF Technologies," in the August 1985 issue of Mobile Radio Technology. This article, written by John Gates, N7BTI, describes the basics of packet radio, its

Gateway

Sept. 4, 1985
Vol. 2, No. 2

The ARRL Packet-Radio Newsletter

TNC 2 HITS THE MARKET

The long-awaited Tucson Amateur Packet Radio (TAPR) TNC 2 went on sale August 19, as scheduled. Volunteers staffed the TAPR office from 9 AM to 9 PM throughout the week, and at the height of the ordering rush, a TNC-2 order was processed every three minutes. The number of phone calls coming into Tucson on the 19th and 20th overloaded the Tucson telephone center several times. Callers had to be both persistent and lucky to place one of the first 300 orders. Those first callers have already received their TNCs. Those that weren't in the first 300 orders will receive their TNCs from a mid-September and an October shipment. TAPR has accepted 832 orders. Packeteers from all over the U.S. and from foreign countries on every continent ordered TNC 2s.

The kits have gone together easily, although IC sockets damaged in shipping and some faulty ICs have hampered the assembly of some TNCs. If you have built a TNC 2 and it passes all tests up to the power-up LED sequence, you may have a bad IC at U13. TAPR has always had a policy of replacing defective parts, and will continue to offer this support. If you suspect that have a bad IC, call the TAPR office and they will send you a replacement 74HC4066. New packaging for the kits should protect IC 'sockets from damage.

Comments on the documentation, which includes an extensive introduct ion to packet operation, have been positive. The software, with a real-time clock and a new, more-informative monitor mode, has also gotten good reviews.

Tom Clark, W3IWI, one of the TNC-2 beta testers, has compiled some notes on operating and customizing the TNC 2. These notes are posted on many packet bulletin board systems (PBBSs) around the country, as well as on some telephone BBSs. Further notes, along with suggestions and comments from kitbuilders will appear in the October issue of the Packet Status Register, quarterly publication of TAPR. If you want immediate up-to-the-minute information on TNC 2 and other developments in packet radio, consider joining the HAMNET (part of the Compuserve on-line information service).

Ed.

220-MHz FREQUENCIES FOR 1200 BIT/S

The ARRL is interested in finding out what frequencies are allocated for or used for 1200-bit/s packet operation on 220 MHz. Art Reese, K9XI, editor of 220 Notes would like publish this information so that his readers will know where to expect packet-radio operation. If you are operating at 1200 bauds on 220 MHz, or if a frequency-coordinating body in your area has allocated a channel for low-speed packet radio, send the information to Gateway.

Ed.

CALL FOR PSR ARTICLES

Submissions for the October issue of the TAPR Packet Status Register (PSR) should reach the editor by September 15. The issue will include TNC 2 news, updates on TAPR projects, and discussions of linking hardware and software. Of course, articles covering other topics, especially articles on the history of packet radio, are welcome.

Editor of the PSR is Gywn Reedy, and his address is:

812 Childers Loop
Brandon, FL 33511.

Via DRNET.

PENNSYLVANIA SITES NEEDED

In late July, packet-radio operators from West Virginia, Ohio and Pennsylvania met in Breezewood, Pennsylvania. The meeting was organized by Gary Hoffmann, AK3P, and Bill Stash, WA3AOQ, and was attended by 14 other packeteers. The discussions at the meeting centered on how to link EASTNET to Pittsburgh, Pensylvania, and then into Ohio.

Several of the operators from Ohio reported that they can regularly work east into Pittsburgh, but there is still no link between Pittsburgh and Harrisburg, the western outpost of EASTNET. Those at the meeting agreed that a digipeater at Blue Knob, Pennsylvania, would be ideal. An amateur-radio voice repeater on this site has very good coverage of the central part of the state. Nobody has been able to get permission to place a digipeater on this site yet.

If you have access to a digipeater site that can put a good signal into both Harrisburg and Pittsburgh, please contact:

Gary Hoffmann, AK3P
1235 Middletown Road
Hummelstown, PA 17036,

or

SOUTHERN CALIFORNIA CLUB ORGANIZES

Packet-radio users in Riverside and San Bernardino counties are organizing the Southern California Amateur Packet Radio Assiciation (SCAPRA). The first Meeting of SCAPRA will be held on August 27, at 7:00 PM, in the lobby-level amphitheater at Loma Linda University Medical Center. A talk-in station will be on the 147.735-MHz LLUARC repeater.

SCAPRA hopes to distribute packet information, coordinate packet activities and promote friendly expansion of packet radio.

For more information, call:

Rod Wertz, WC6T
714-796-2883

From NI6A, HAMNET.

REPRODUCTION OF GATEWAY MATERIAL

Material may be exerpted from Gateway without prior permission, provided that the original contributor is credited and Gateway is identified as the source.

Gateway: The ARRL Packet Radio Newsletter is published by the

American Radio Relay League
225 Main Street
Newington, CT 06111

Jeffrey W. Ward, K8KA
Editor

Larry E. Price, W4RA
President

David Sumner, K1ZZ
Executive Vice President

Subscriptions are available to ARRL members and nonmembers, at the following rates:

For 25 issues of Gateway:
- In the U.S.
  - ARRL Member $6.00
  - Nonmember $9.00
- In U.S., Canada, and Mexico by First Class Mail
  - ARRL Member $11.00
  - Nonmember $14.00
- Elsewhere by Airmail
  - ARRL Member $14.00
  - Nonmember $17.00.

Prices are subject to change without notice. Remittance must be in U.S. funds, and checks must be drawn on a bank in the U.S.

Gateway: The ARRL
Packet Radio Newsletter
American Radio Relay League
225 Main St,
Newington, CT 06111

Nonprofit Organization
U.S. Postage
PAID
Hartford, Conn.
Permit No. 2929

will be held on the campus of Georgia Technical University, in Atlanta. Technical sessions will run Saturday from 8:30 to 5:00 and on Sunday from 8:30 to 12:O0. There will be an informal get-together on Friday night and an organized dinner on Saturday.

Interested packeteers are invited to make formal or informal presentations on any topic concerning packet radio. If you want to participate, send your name, address, phone number, amount of time needed, a list of audio/visual equipment needed and a brief abstract of your topic to:

Bill Crews, WB2CPV
1421 Hampton Ridge Road
Norcross, GA 30093

From N4CI.

NEW PACKET GROUP IN SOUTH CAROLINA

On Saturday August 10, a group of 10 amateurs met in Columbia, South Carolina to organize a statewide group for hams interested in digital communications. The name for the group is South Carolina Amateur Digital Society or SCARDS. The main purpose of SCARDS is to provide a forum in which amateur radio digital communicators can get acquainted with each other and exchange informat ion. It will also promote the orderly growth of packet radio and other forms of digital communications in South Carolina. SCARDS will encompass all forms of digital communication, not limiting itself to packet radio. The club will publish a newsletter every 2 months, beginning in September. Items for the newsletter should be sent to editor Al Nelson, KA4YEA.

This item came by way of the GRAPEVINE, newsletter of the Georgia Radio Amateur Packet Enthusiasts. The GRAPEVINE also reports that on the weekend of August 17, a team of hams installed the W4FX-1 digipeater on Sassafras Mountain, the highest point in South Carolina. This digipeater should help fill the gap between EASTNET and SOUTHNET, as it sits midway between packet-radio activity centers in North Carolina and Georgia.

Via W4FX, DRNET.

MULTIPLE CONNECTIONS

If you are interested in experimenting with multiple simultaneous connections from the same TNC, you will be interested in these developments. Multiconnect firmware written by WA8DED is already in use at several sites in California. Version 0.9 of this software (for the TAPR TNC 1 and compatibles) is now available. This software offers the following features: up to four simultaneous connections possible, easy-to-read monitor mode, on-screen calibration, digipeats AX.25 Version 2 frames, separate status display for each connection and a special “host mode” for computer interfaces.

Harold Price, NK6K, also has multiconnect software in the beta-test stage. This is part of the upcoming TAPR TNC 1 Version 4.0 software. Owners of the TAPR TNC 2 will not be left out; Howard Goldstein, N2WX, author of the TNC-2 software is testing multiconnect routines for that TNC.

The availability of this software should spur development of multiuser bulletin boards and gateways. It will also provide a basis for experimentation with networking protocols. We are interested in hearing from anyone developing applications that will take advantage of multiple level-2 connections.

Via HAMNET, DRNET.

NEW PACKET CENSUS

The following is both a report and a request from Harold Price, NK6K:

“While southern California isn't always a good place from which to judge how things are going in the rest of the world, it is interesting to note how quickly packet is growing around here. On our Monday-night nets (both voice and digital), we have been getting several new users each week. Most of them are running the Heath HD-4040 TNC.

"TNCs seem to be selling very well. A call to a LA radio store revealed a backlog of orders for AEA TNCs. Heath is sold out of their latest product ion run, and they are again back ordered. Based on the production schedules the Heath people were talking about at Dayton, this means that they've sold around 1500 TNCs.

“What is the new-user activity in the rest of North America? I would like to do another packet census, so send me any information that you have. Send reports to NK6K @ KD6SQ, via any WØRLI MailBox that can reach an HF Gateway.”

From NK6K.

WESTNET PROJECTS

Looking for something to do this winter? This item, which originated at the WA6OSA Packet Mailbox, details the projects underway in WESTNET:

- Tom King, KA6SOX, is developing a 9600-bit/s FSK modem using K9NG technology adapted for use with Midland 220-MHz radios. Tom is also developing modifications for the Midland radios to make them more suitable for WESTNET backbone use.

- Greg Pierro, WA6RWN, and Jerry Brayton, WB6AIE, are developing a CMOS Z80-based processor board for use in remote mountaintop environments.

- Orv Beach, WB6WEY, is enhancing the KE3Z multiport digipeater software and the FAD HDLC daughter board for WESTNET backbone use.

- Andy Cromarty, N6JLJ, is drafting a proposed functional specification for WESTNET networking software.

- Mike Busch, W6IXU, is developing new mailbox software which uses the WA8DED multiconnect firmware. It will support multiple simultaneous connections and will soon include automatic store and forward with other mailboxes. The WA6OSA mailbox will be converting to the new software as soon as it is available.

Via HAMNET.

and a big gap between **EASTNET** and SOUTHNET. To the west, there is significant activity from Pittsburgh west to Finley, Columbus, Indianapolis and into Chicago. We would like to investigate routes through West Virginia and central Pennsylvania. Anybody with knowledge and ideas about these routes should be prepared to make a presentation. How far away is the Golden Packet award?

0 Hardware and new-user **discussions.** The newest packet radio hardware, the TNC 2 (complete with **ALJ-1000),** the Kantronics Packet Communicator and the AEA **PK-64,** will be discussed. Bring your new application6 software, especially terminal programs for common personal computers. Time will be allocated for discussing the problems that new user6 face when coming on the air.

0 Frequency-coordination matters,boththose within the packet community and external matter6 like local repeater council6 and FCC actions.

0 Integration of packet radio into other aspects of Amateur Radio, particularly the National Traffic System, emergency **communicat** ion6 and disaster relief. The recent use of packet systems in the California fires has raised questions about emergency communication and emergency preparedness.

0 Packet bulletin board system **(PBBS)** discussion. What'6 happening on the PBBS front? How do I get a PBBS on the air in my area? How will **PBBSs** evolve in the next few years? How do **RCP/M systems,** UNIX host6 and HF Gateways fit into the grand scheme? This will also be a discussion of PBBS coordination.

0 DX packet radio, including HF gateways, Oscar 10, PACSAT, **JAS 1** and the possible **WA4SIR** Space Shuttle packet-radio experiment.

As at past **MAPRC** meetings, these **discussions** will be in "town-meeting" format; everyone will be encouraged to speak his or her piece. Please let us know if you are going to be able to come to the meeting. Also, contact us if there are items that have been omitted from the discussion topics listed above.

From W3IWI.

VADCG DEVELOPMENTS

Although we don't hear about them often, there are quite a few VADCG **TNCs** in operation. The latest VADCG news comes from Alan Mar, **VE7DPM:**

The Sydney (Australia) Amateur Digital Communication6 Group **(SADCG)** has taken the VADCG V-2 protocol and the AMRAD implementation of AX.25 for the VADCG TNC and merged them. This allow6 one VADCG TNC to run both protocols. It is now possible to choose either V-2 or AX.25 from the TNC menu. Since more memory is required when you have both protocol6 in **one** TNC, you must modify the TNC to accept larger EPROMs. Alan does not know if the Australian6 have a version of this software that runs on a VADCG TNC with the AMRAD daughter board.

An interesting feature of this software is the inclusion of a standard TNC-to-user interface. This interface, based on international standards X.3 and X.28, define6 TNC operating parameters and provides a protocol for displaying and altering them. The ARRL Ad Hoc Committee on Digital Communication6 is investigating the adoption of this protocol as an amateur standard. (See the proceedings of the 4th **ARRL** Amateur Radio Computer Networking Conference for further details.) VADCG devotee6 have developed a **CP/M** program that uses the **X.3/X.28** interface to transfer binary files. The file-transfer protocol is TNC-independent, and details should be available **soon.**

Via HAMNET, VE7DPM.

WISCONSIN NEWS

John Corstvet, **WA9SOU,** from Madison, Wisconsin provides this activity report:

"At the July meeting of the Wisconsin Amateur Packet Radio Association (WAPRA) the club decided to move Wisconsin packet activity from 144.950 MHz to the national packet frequency, 145.010 MHz. Most station6 will have already switched frequencies. We are working on a digipeater at North Freedom, Wisconsin, which will link with stations in Madison, Baraboo and Portage. The North Freedom system should be on the air by early September. We are also making plans for a path from North Freedom to La Crosse, from La Crosse to Rochester, Minnesota and then north to the Minneapolis/St. Paul, Minnesota area. If you are interested in more information on WAPRA, their address is:

WAPRA
P.O. Box 1215
Fon du Lac, WI 54935.

Membership dues include a l-year subscription to Badger State Smoke Signals, which carries the WAPRA newsletter."

From HAMNET, WA9SOU.

BARE PC BOARDS AVAILABLE

Applied Digital Technology **(ADT),** of Oxnard, California, is selling bare pc boards for the TAPR TNC 1. The boards, produced from the TAPR artwork, are faithful copies of TNC 1. ADT is also selling the latest TAPR firmware, reproductions of the TAPR manual and some of the hard-to-get TNC parts. According to ADT, the boards are high-quality glass epoxy, solder masked and silkscreened with component locations. The company may produce pc board6 for TNC accessories. For further information, contact:

Applied Digital Technologies
2056 E. Sutter Place
Oxnard, CA 63033
805-488-5575.

Via DRNET.

SOUTHNET CONFERENCE II

The Second **SOUTHNET** Packet-Radio Conference will be held on November 23 and 24. The conference, hosted by the Georgia Tech Amateur Radio Club,

Gateway

Vol. 2, No. 1
Aug. 20, 1085

The ARRL Packet-Radio Newsletter

HAWAIIAN ISLANDS LINKED

The Maui County ARES Bulletin reports that Maui (the largest of the Hawaiian-islands) and the island of Kauai are now linked by two-meter packet radio. Bill Baisley, KH6S, and Army Curtis, AH6P, made the first inter-island connection, using a digipeater on Mauna Loa. The Mauna Loa digipeater is on 145.05 MHz, operating as AH6P-1.

From WESTLINK.

OHIO LINKING

Thomas Kryza, KB8CI, reports another packet-radio first:

"Perhaps it wasn't an earth-shaking event, but nevertheless, July 14 marked a milestone for Ohio packeteers. Tom Kryza, KB8CI, in Cleveland, and Reggie Brown, KI4UN, in Florence Kentucky (just across the Ohio River from Cincinnati) made the first Cleveland-to-Cincinnati packet-radio QSO. The contact over the 292-mile path lasted about a half an hour. Two digipeaters were used: W8AC in Chardon, Ohio and AD8I in Circleville, Ohio.

"The more than 30 packeteers around Cleveland and Akron are now connecting on a regular basis with Detroit, Youngstown, Pittsburgh and London (Ontario). We are looking for stations in Sandusky and Mansfield to increase the reliability of paths to Toledo and Columbus."

From KB8CI.

IOWA FREQUENCY COORDINATION

From the minutes of the May meeting of the Iowa Repeater Council:

"A motion was made, seconded, and passed that 145.01, 145.03, 145.05, 145.07 and 145.09 MHz be set aside for packet radio. A motion was also made, seconded and passed to change the 220 bandplan to accommodate high-speed data links.

"It was suggested that since we have decided to stay with the 15/30-kHz plan, that we also change the packet-radio subband from the 20-kHz plan to the 15/30-kHz plan so that the entire 2-meter bandplan is consistent. It was decided to study the idea and delay decision until a later date."

From the IRC.

MID-ATLANTIC MEETING

On September 7, the Mid-Atlantic Packet Radio Club (MAPRC) will hold a meeting in conjunction with the Gaithersburg Hamfest. MAPRC is primarily intended to act as a unified voice for packet radio in northern Virginia, Maryland, Delaware, eastern Pennsylvania and southern New Jersey, but anyone who is interested in packet radio is invited to attend.

Saturday was selected to allow a full afternoon of packet-radio discussions without interruption from activities at the Hamfest. The meeting will start around 14:00 EDT, with dinner planned for 18:O0. Meeting facilities will be available starting about 13:00 for informal discussions, disk swapping, etc.

The meeting will be at the Baldwin Community Hall in Savage, Maryland. This is within a mile of the intersection of US 1 and MD 32, and it is convenient for travellers coming in on on I 95, US 29 or the Baltimore/Washington Parkway. Detailed directions will be posted on major EASTNET packet bulletin board systems (PBBSs), addressed to the dummy address "MAPRC". Talk-in stations will be on the Maryland FM Association 146.76/.16-MHz repeater and the Columbia ARA 147.135/.735-Mhz machine.

Some of the local packeteers have offered to put up those travelling from afar on modest budgets. These "luxury" accommodation6 range from tents and RVs to floor space suitable for sleeping bags. To make arrangements, please send your lodging requirement6 through EASTNET PBBSs to the address BEDS @ W3IWI.

The meeting will address the following topics:

- MAPRC organization. It is time for MAPRC to become a "real" entity.

- The expansion of EASTNET in the MAPRC area. This will include a discussion of 9600-bit/s linking, dual-port digipeaters and the need for fund6 to get an enhanced network on the air. What about the proposed TAPR network controller? What is happening with level-3 linking protocols? Will TCP/IP or VC get on the air first? Just what are level 3, TCP/IP and VC, anyway?

- Expansion of EASTNET north, south and west. From Harrisburg north through Williamsport, State College, Allentown, Scranton, Wilkes Barre and into western New York, the links are sparse. Stations in that area are now best reached through southeastern New York. South from Washington there is sparse activity through southern Virginia

REPRODUCTION OF GATEWAY MATERIAL

Gateway: The ARRL Packet- Radip Newsletter is published by the

American Radio Relay League
225 Main Street
Newington, CT 06111

Jeffrey W. Ward, K8KA
Editor

Larry E. Price, W4RA
President

David Sumner, K1ZZ
Executive Vice President

Subscriptions are available to ARRL members and nonmembers, at the following rates:

For 25 issues of Gateway:
- In the U.S.
  - ARRL Member $6.00
  - Nonmember $9.00
- In U.S., Canada, and Mexico by First Class Mail
  - ARRL Member $11.00
  - Nonmember $14.00
- Elsewhere by Airmail
  - ARRL Member $14.00
  - Nonmember $17.00.

Prices are subject to change without notice. Remittance must be in U.S. funds, and checks must be drawn on a bank in the U.S.

Gateway: The ARRL
Packet Radio Newsletter
American Radio Relay League
225 Main St.
Newington, CT 06111

Nonprofit Organization
U.S. Postage
PAID
Hartford, Conn.
Permit No. 2929

BULLETIN BOARDS

CLUBS

COMMERCIAL PRODUCTS

EASTNET

FCC ACTIONS

HF PACKET

IBM PC

INTERNATIONAL

MIDWEST

MODEMS

MOUNTAIN STATES

NETWORK COORDINATION

NETWORKING

Gateway

Vol. 1, No. 25
Aug. 6, 1985

The ARRL Packet-Radio Newsletter

TAPR TNC 2 AVAILABLE AUGUST 19

Tucson Amateur Packet Radio ha8 announced that its new TNC 2 will be available beginning August 19. No order8 are being taken before the 19th. Order8 will be taken by telephone between 9 AM and 9 PM PDT, August 19 through 23, and during regular business hour8 thereafter. There are **280** TNC **28** on hand at **TAPR,** and these will be shipped immediately. Another 300 **TNCs** should be ready to ship sometime in mid-September. When you call to order a **TNC,** you will be told whether your8 is to be part of the first batch or the second batch. The price of the TNC 2, including shipping, handling and credit-card charge8 will be **$200.85.** The TAPR phone number is **602-746-1166.** Congratulations to the TNC-2 volunteers for completing a **signif icant** packet-radio project.

From DRNET.

IOWA ADDRESS CHANGE

The Central Iowa Technical Society **(CITS)** has a new president and a new address. Contact CITS at:

Richard **E . Amundson, WAØFJS**
President CITS
4621 SW 2nd St.
Des Moines, IA **50315**

From DRNET.

NOTES FROM GLB

Ken Burton, **N9CVV,** reports the following developments at GLB. GLB will be coming out with two new **models,** the **PKlL** and the PK2. The **PKlL** will be the same as the the **PKl,** except that it will require just under 40 **mA** at 12 **V.** The PK2 is supposed to be entirely new, with some new features. They may also be offering a complete packet station: receiver, transmitter, amplifier and TNC all mounted in a **19" chassis.** Simply add a power supply and you have a rack-mountable digipeater. Ken saw the prototype of a complete **220-MHz** network controller. It will run 9600 **bits/s** and include transmitter, radio/modem and network control computer. GLB would 1 ike to hear what the amateur community wants designed into this controller. Write to Ed Jackson or Gil Boelke at:

GLB Electronics
151 Commerce Parkway
Buffalo, NY 14224.

Via **DRNET.**

INDEX TO VOLUME 1

This is the final issue of Volume **1** of Gateway. The first issue was printed on August 14, 1984, and in the year since then, we have signed up 730 subscribers. A few hundred League officials and Special Service Clubs receive free copies of the newsletter, and several thousand packet operators and bulletin-board callers get the "electronic edition*' of Gateway. Thank you for your support.

A subject index to Gateway Volume **1** follows. The categories are broad, but you should be able to find what you are looking for. All entries under **"Clubs"** include club addresses. Emergency, NTS and ARES items have been grouped under "Public Service." Back issues of Gateway are **$0.50** from the ARRL Publications Sales Department.

Entries are in the form:

Title, **V:I,** p **N.**

V is the volume, **I** is the issue and N is the page number. Subject headings are in bold face type.

MINICOMPUTER ON PACKET

St. Louis Amateur Packet Radio (SLAPR) has acquired a DEC PDP-8 minicomputer. They would like to put the machine to some practical use on the local packet network. Because there are already several mailboxes on the St. Louis-area network, the club is interested in using the PDP-8 to store and distribute large files like Gateway. If you have any expertise that you might donate to this SLAPR project, contact

Pete Eaton, WB9FLW
35 Norspur, Route 4
Edwardsville, IL 62025.

VIA DRNET.

REPRODUCTION OF GATEWAY MATERIAL

Material may be exerpted from Gateway without prior permission, provided that the original contributor is credited and Gateway is identified as the source.

Gateway: The ARRL Packet Radio Newsletter is published by the

American Radio Relay League
225 Main Street
Newington, CT 0611.1

Jeffrey W. Ward, K8KA
Editor

Larry E. Price, W4RA
President

David Sumner, K1ZZ
Executive Vice President

Subscriptions are available to ARRL members and nonmembers, at the following rates:

For 25 issues of Gateway:
In the U.S.
ARRL Member $6.00
Nonmember $9.00
In U.S., Canada, and Mexico by First Class Mail
ARRL Member $11.00
Nonmember $14.00
Elsewhere by Airmail
ARRL Member $14.00
Nonmember $17.00.

Prices are subject to change without notice. Remittance must be in U.S. funds, and checks must be drawn on a bank in the U.S.

Gateway: The ARRL
Packet Radio Newsletter
American Radio Relay League
225 Main St.
Newington, CT 06111

Nonprofit Organization
U.S. Postage
PAID
Hartford, Conn.
Permit No. 2929

Mt. Airy, Maryland, K3VPZ near Baltimore and N8FJB in Harper's Ferry, West Virginia. Also, Bob Bruninga, WB4APR, has his Commodore VHF-to-30-meter gateway software converted to support commands and mail forwarding like a WØRLI system. So, the Baltimore/Washington network has the following major stations on 145.05 MHz: N8FJB, W3TMZ, K3VPZ and WB4APR-5. KS3Q scans 145.01, 145.05 and 145.09 MHz.

VIA EASTNET.

NEWS FROM SOFTNET

The Swedish SOFTNET User Group (SUG), in Linkoping, Sweden, has had its third workshop. SOFTNET is an experimental packet-radio system that uses distributed processing to provide network services. The SOFTNET nodes run at 100 Kbits/s and are remotely programmed in a FORTH-like computer language. [For more background on SOFTNET, see Gateway Number 5.1 The following report on the SOFTNET workshop is from the newsletter SOFTNET News.

The event was attended by about 50 SUG members and other participants from Sweden and Denmark. The first session started with introductory talks on the SOFTNET concept. These talks were followed by a hardware session which was dominated by Ingvars, SMSLRF, who presented the latest news on the Softnet radio transceiver [100-kbit/s modem]. While the transceiver has been available for about six months, minor modifications to improve the performance are continuously being made. During the lunch break, there were demonstrations of SOFTNET nodes and connections to several main-frame computers.

The first session after lunch was devoted to SOFTNET programming. Fritte, SM5PSL, talked about activities of the SOFTNET Specification Group and presented some of the group's proposals. Peter, SM5PQZ gave a talk on the implementation of an algorithm for distribution of system software through intelligent network flooding.

An interesting talk on some of the commercially available packet radio systems was given by a member of the Swedish Telecommunications Administration. SOFTNET-like systems are of great interest to the Administration because they can solve data communication problems for fixed and temporary networks in difficult terrain.

The next session was on packet radio in satellite systems. Sigge, SM5KUX, presented some of the plans on SWASAT, a Swedish Amateur Radio Satellite which is to carry advanced, high-speed packet-radio experiments. An interesting paper on the design of a doppler-shift adaptive modem for the IC-251/451 was also presented.

The workshop concluded with a panel discussion chaired by Carl-Lennary, SM5ELD. Among the biggest issues of the discussion was the negative view of digital communications and packet radio taken by the HF Working Group of the IARU. The participants urged the spreading of more information to bring on a more positive attitude towards packet radio and experimental activities in general.

VIA SOFTNET NEWS, SM5PSL.

QSL TO HELP DRNET

One of the constant sources of Gateway news is DRNET, a computerized conference that is part of the New Jersey Institute of Technology's, Electronic Information Exchange System (NJIT EIES). NJIT is interested in knowing how many people are served by DRNET. As Gateway readers, you all receive regular DRNET information. Please send a post card to

CCCC @ NJIT
Tom Moulton, W2VY
323 King Blvd
Newark, NJ 07102.

Tom is the NJIT "sponsor" of DRNET, and he would like to be able to show that the 30 DRNET accounts are actually serving several thousand information-starved packeteers. He is also interested in how far messages from DRNET get distributed. On your card to Tom, tell him your name, call, location and packet setup, how you got this message, how often you receive messages that originate on DRNET, the name of your local packet group and the number of members in your group. The few minutes that you take to do this will help keep a vital source of information active.

VIA DRNET.

COHERENT DEMODULATION?

Fred Weidenhammer, W4SDL/2 writes that he is looking into the application of the techniques involved in coherent CW·(CCW) to HF packet demodulation. If you are interested in this, contact Fred and experiment. His address is

322 Blacksmith Road
Levittown, NY 11756.

FROM W4SDL.

600-Hz SHIFT FOR HF PACKET?

While all of the stations using packet radio on HF are using 200-Hz shift at 300 bauds, there has been a lot of discussion of eventually using wider shifts. To start some public dialog and experimentation on this matter, here is a letter sent to Gateway by Dr. Alan Chandler, K6RFK.

"The 200-Hz shift that is becoming the defacto standard for 300-baud HF packet has a substantial deficiency when on an ionospheric path. Due to differential path length, both frequencies may be present simultaneously at the receiver and will generate a beat frequency. If the beat frequency is greater than the highest desired harmonic of the data, it may be filtered out and will not be a problem. If it is lower, the beat will be treated as data by the TU and will generate errors. In the case of 300-baud data, the fundamental is 150 Hz, and the minimum 3rd harmonic should be passed. This gives a minimum shift of 450 Hz, and to allow economical filters, the minimum shift should be 600 Hz, The TAPR and TAPR equivalent modems will receive 600 HZ shift without modification and they can easily be adjusted to transmit 600-Hz shift tones (1400 and 2000 Hz)."

FROM AEA.

performance left much to be desired...Only one QSO could be managed. But that produced another 100 points. On 145.01 MHz, where the packet action took place, the racket has to be heard to be believed. At times it sounded like one long 'brrrap' as one packet followed another the instant the frequency was clear. It was a good demonstration of how packet allows a lot of stations to work on the same channel concurrently without QRM.

"I 'printed' stations from all over Dallas and Tarrant counties. ..Many stations were extending their range by using the 'digipeater' capability of the AX.25 packet protocol.

"Field Day has shown that packet can add another dimension to traditional contest modes, as well as proving that you can take reasonably delicate home computer equipment and set it up in the field under adverse conditions and make it play...Packet radio is really out there, and it is beginning to 'intrude' into areas of ham radio that formerly were the exclusive domain of CW or phone."

David Cheek, WASMWD, of the Texas Packet Radio Society, operated at the K5QHD Field Day site in Garland, Texas .

"More than twenty three packet stations in the North Texas Section participated in the Field Day. About half were operating in the field, on emergency power. At K5QHD in Garland, we made 20 contacts, while two stations in Ft. Worth, made 23 each. At least two other stations went over 15 contacts, including W5CR in Athens. The best DX on two meters was with Salado on packet. Houston packet stations made lots of contacts with the help of the WD5GAZ digipeater.

"We learned a few lessons: First, make sure all of your packet equipment is ready to operate in the field. Try to have 25 watts or more and an antenna at least 30 feet high. Second, finding a route to a station can be difficult, as is learning the call of the distant station you want to connect to. This can be handled by some limited beaconing, with each station continuously monitoring to gather new information about who is operating were. This can all also be mapped out in advance. Third, excessive beacons can seriously slow down traffic on the frequency and should never be any more frequent than once every ten minutes. Beacons must include useful information, such as, 'reach me through KC5LW-1, WB5QNG and WA5MWD'. Fourth, the frequency must be kept as clear of other traffic as possible. This is no different from any other emergency operation. Finally, in a real emergency operation, two people are needed for each packet station. One will operate the transmitter while the other types traffic into a message computer. Only a station handling very little traffic -- less than ten ARRL radiograms per hour -- can be operated by one person."

There are two Field Day reports in the June/July issue of the NEPRA PacketEar, newsletter of the New England-et Radio Association. John Langner,. WB2OSZ, operated with the Billerica (Massachusetts) Amateur Radio Society. They only made a few contacts, but did use packet radio to receive the official Field Day bulletin and send a message to the Section Manager. The packet operators explained packet to those who had never seen it and may have inducted a few new packeteers.

Robert Gettys, N1BRM, was the sole packet operator at the Framingham (Massachusetts) Amateur Radio Association Field Day. Robert found that the lack of a second operator familiar with packet kept the station from being operated for the full contest period. He "found that you can't teach someone who doesn't understand the concepts of computers in general how to run a packet station and 'search' for possible contacts...or even enough [to conduct] a simple contact." Robert made 21 contacts, and he was able to send and receive Field Day messages from the K1BC PBBS.

Robert concludes that we need to find a recognized procedure for soliciting contacts (calling CQ), and he also notes that beacons and PBBSs contributed to severe channel congestion. His article ends with a few questions that the packet-radio and contesting communities should begin to think about:

0 Should packet be used in contests at all?
0 If so, in which contest(s)?
0 What rules should be specified concerning digipeaters, beacons and frequencies?
0 Should contacts with PBBS or stations under automatic control count?

Robert feels that because Field Day is as much a test of emergency communication as it is a contest, packet radio is probably appropriate for Field Day. He is less sure about other contests.

If you would like to put forth answers to any of Robert's questions, send a letter to Gateway and to the members of the ARRL Contest Advisory Committee.

CHAWED RAG, DRNET, PA-.

EASTNET NOTES

Gary Hoffmann, AK3P, in Hummelstown, Pennsylvania, has his WØRLI MailBox system running on 145.05 MHz as well as on 145.01 MHz. This allows users of the 145.05 MHz digipeater in York to access his system. Gary is using two radios in parallel to implement his dual-frequency node. He hopes to install a true multiport digipeater in the future. Gary is also planning to add 8n HF port to his MailBox, with antennas beamed toward Europe.

Tom Clark, W3IWI, is also operating a MailBox with both 145.05-MHz and 145.01-MHz inputs. Tom, however, is using the WORLI Gateway software to connect the two frequencies. To reduce the congestion that is beginning to bother everyone on 145.01 MHz, all local users in Baltimore and Washington D.C. have been encouraged to use the 145.05-MHz port. This helps keep 145.01 MHz open for EASTNET traffic entering and leaving the Washington area. The 145.05-MHz input is served by the W3GXT-5 (Baltimore) and W3VD-5 (Laurel/Columbia) digipeaters. The novel thing about this installation is that the radios for the two ports are combined using an RF-hybrid combiner to drive a single power amplifier and antenna.

Tom sends these other notes: WORLI MailBox systems are beginning to be used as personal mailboxes. Three new digipeaters on 145.05 MHz are W3TMZ in

Gateway

Vol. 1, No. 24
July 23, 1985

The ARRL Packet-Radio Newsletter

FIGHTING FIRES WITH PACKET RADIO

This report on packet-radio communications during the Lexington and Santa Cruz Mountain (California) wildfire was filed by Ken Chong, **WB6MLC,** and edited by **Hank** Magnuski, **KA6M.**

"During the afternoon of Sunday, July 7, reports were received by the local Amateur Radio Emergency Service (ARES) District Emergency Coordinator (DEC) that a new fire had been discovered in central Santa Clara County near the Lexington Reservoir, about 60 miles south of San Francisco.

"Since the 4th of July the local ARES 'Strike Team' had been assembled and put on alert. This particular amateur radio team, unlike previous ones' was equipped with a complete packet radio station. Late that Sunday, the team was called out to set up operations at the Alma Fire Station to handle communications concerning resources, tactical messages, fire updates and evacuation notices .

"Throughout the day and further into the week, Fire Status Updates were sent out over packet when the voice nets were tied up. Once, when the phone lines were busy, packet radio got through to **KA6YRK** at another fire camp in San **Luis Obispo,** in Southern California. Many operators who had never operated packet before were literally baptized by fire!

"On Tuesday, July 9, with the fire leaping rapidly from hill to hill, the main base camps and staging areas were moved to Vasona Park in Los **Gatos.** The towns of Los Gatos and Saratoga were severely threatened and many held their breath, hoping that evacuation orders would not be issued. At **'Vasona** Corn,' where the future amateur operations were to originate, a well-equipped van from the Palo Alto Red Cross was set up with packet equipment and an IBM PC programmed for use as a terminal. Two other radios for the voice nets were also in the van. Again, packet radio continued to function when the voice nets were busy.

"One glaring problem, receiver desense, reared its ugly head during the operations. The local mountaintop digipeater was also malfunctioning, and two meter full-duplex repeaters were pressed into service. The retry rate increased so much because of the **intermod** products and **desense** that the packet technical crew transferred all digital traffic to full-duplex voice repeaters on **440** MHz. Finally, with the packet net secured on 440 MHz, all relays of fire updates to the different base camps continued without interference from the nearby two-meter radios.

"Thanks to the hard work and contributions of many programmers, the packet operation at the main fire camp solved one more disturbing problem: The large pool of fresh operators had little or no experience with **TNCs. Most** had barely heard of TAPR! The IBM PC was cleverly programmed to use its function keys to connect, disconnect and send various 'canned' messages. By Thursday, July 11, the whole operation was almost foolproof. At last an untrained operator could sit down and keep the net going.

'Packet radio did an outstanding job in proving to the fire officials its usefulness in an emergency and at the fire camps. The initial strike team of Frank Kibbish, **WB6MRQ;** Kit Blanke, **WA6PWW** and Bob **Tarone, WA6ZBX** and the packet setup crew of David Palmer, **N6KL** and **WB6MLC** were pleased with what evolved during the course of the emergency. When the smoke clears, a more detailed report will be filed.

VIA DRNET

ANOTHER FIRE REPORT

Paul Hansen, KA6UPD, sent in a newspaper article detailing some of his activities as a packet-radio operator assisting California firefighters. The article appeared on the front page of the July 14 edition of the Thousand Oaks,' California. News Chronicle, and it carried a photo of **Paul and** Jerry Boone, emergency coordinator for Ventura County ARES. The article gives some description of packet radio and the of work done by ARES members in the California disasters. **Several** paragraph6 are devoted to the reasons that Amateur Radio operator6 get involved in public service. Paul is quoted, saying, "We have an obligation to give public service to pay back for the right to use the frequency space we have." Ray Volkmar, leader of the **Conejo** Valley Disaster Action Team, says that public service stems from "a love of our hobby and a love of being able to do something for someone else."

VIA KA6UPD.

FIELD DAY REPORTS

It looks like a lot of stations took advantage of the **100-point** packet-radio bonus this Field Day. The following reports detail some successes and some lessons learned.

In The Chawed Raq, neweletter of the **Richardson (Texas) Wireless** Klub, George Baker, **W5YR, reports** that "unfortunately, the packet **station**

TAPR PROJECT STATUS

The TAPR TNC 2 is in the "final stages" of beta testing, with the hardware completely debugged and the final version of the software distributed to the dozen beta testers. The manual, one of the most demanding aspects of the project, is almost complete. It looks like initial estimates of "late summer or early fall" for availability of the first 300 TNCs were fairly accurate. The process of announcing the availability of the TNC 2 and then taking orders for it will be tightly controlled. When TAPR is ready to take orders, electronic messages will be posted on DRNET and Compuserve HAMNET. Orders will be taken over the telephone, and only one order per person and one order per phone call will be accepted. No COD or purchase orders will be accepted. Once the first 300 TNCs have sold out, there will be NO WAITING LIST. These rules will not be bent.

The TAPR PC board for the K9NG 9600-bit/s modem is also just about through beta testing. Several of the boards in operation at beta sites.

Ed.

REPRODUCTION OF GATEWAY MATERIAL

Material may be exerpted from Gateway without prior permission, provided that the original contributor is credited and Gateway is identified as the source.

Gateway: The ARRL Packet Radio Newsletter is published by the

American Radio Relay League
225 Main Street
Newington, CT 06111

Jeffrey W. Ward, K8KA
Editor

Larry E. Price, W4RA
President

David Sumner, K1ZZ
Executive Vice President

Subscriptions are available to ARRL members and nonmembers at the following rates:

For 25 issues of Gateway:
In the U.S.
ARRL Member $6.00
Nonmember $9.00
In U.S., Canada, and Mexico by First Class Mail
ARRL Member $11.00
Nonmember $14.00
Elsewhere by Airmail
ARRL Member $14.00
Nonmember $17.00.

Prices are subject to change without notice. Remittance must be in U.S. funds, and checks must be drawn on a bank in the U.S.

Gateway: The ARRL
Packet Radio Newsletter
American Radio Relay League
225 Main St.
Newington, CT 06111

Nonprofit Organization
U.S. Postage
PAID
Hartford, Conn.
Permit No. 2929

This would require a control operator to examine each packet at a rate of 600 words per minute to keep up with the throughput capability of such a repeater. Furthermore, speeds higher than 1200 words per minute are authorized, and technology is advancing toward higher and higher speeds."

After this, the comments address the common practice of using automatically controlled voice repeaters to pass third party traffic. In conclusion, the ARRL says:

"We urge the Commission to modify its proposal so as not to impose a blanket prohibition of third-party traffic on stations operating under automatic control. We believe that the best way to achieve the goal of promoting the developing technologies in digital communications is to leave section 97.79 intact with the possible exception of clarifying section 97.79(d) by [adding the following text]:

> 'Participation means the origination, introduction or reintroduction of a communication into the amateur radio medium by a third party.'

"The proposed addition of this language to section 97.79(d) has the added benefit of clarifying that a control operator is required during operation of simplex autopatches.

'It is further urged that the proposed Rule sections 97.80(c) and 97.114(d) not be adopted, and that the Commission instead adopt the wording of 97.80, as originally proposed by the League, as follows:

"97.80 Automatic Control of Digital Communications.

> 'Amateur Radio Stations may be operated under automatic control on frequencies above 30 MHz when utilizing digital communications pursuant to section 97.69, provided that the control functions include (i) adequate provision for detection of transmitter malfunction and discontinuance of transmitter operation in the event such malfunction is detected; (ii) devices installed and procedures implemented to ensure compliance with the rules when a duty control operator is not present at a control point of the stat ion. Upon notification by the Commission of improper operation of a station under automatic control, operation under automatic control shall be immediately discontinued until all deficiencies have been corrected.

'THEREFORE, the foregoing considered, the League recommends that the Commission extend automatic control to amateur digital communications above 30 MHz. The League also requests that the proposed blanket prohibition of third-party traffic transmit ted on automatically-controlled stations not be adopted, and it urges that section 97 be amended by the adoption of section 97.80 as originally proposed by the League."

Ed.

NORTHWEST NEWS

The following update on the Northwest Amateur Packet Radio Association (NAPRA) comes from John Hayes, KD7UW, NAPRA's Technical Vice President.

"There is currently a network operating in Seattle on 145.010 MHz. In order to reduce the frequency of collisions and bring some organization to the frequency, use of the WN7ANK-5 digipeater is mandatory. Stations that can connect without the digipeater are encouraged to move to 145.050 MHz.

"Northern Idaho and eastern Washington are now also served by a 145.010-MHz digipeater, N7BI-4 on Mica Peak. Again, the suggested simplex channel is 145.050 MHz. The packeteers responsible for the N7BI digipeater are affiliated with NAPRA and work with us on network coordination. To facilitate this, we hold an HF net on 3885 Khz, Thursday at 8 P.M. Pacific Time.

"Doug Lux, WB6VAC, has been advocating the use of a 'channel-busy' signal to reduce the number of collisions at digipeater receivers. The packet repeater with a busy signal will, like a duplex repeater, use two frequencies. Whenever it is receiving a packet, it will transmit a busy signal on its transmit frequency. This signal will be recognized by stations using the repeater, and they will not transmit when they hear the busy signal. Once a packet has been completely received by the repeater, it will be regenerated (as in a digipeater) and retransmitted. The repeater with busy signal is addressed just like a digipeater. This addressing and the regeneration of packets before they are repeated may make the proposed system superior to a standard duplex repeater. WB6VAC will be experimenting to find out if the system works well.

"We are working on expanding our network into Oregon, which is where we will meet WESTNET. A site in Longview, Washington, has been located, and we are working to put in a digipeater there. Longview 'sees' Portland, Oregon. The Utah Packet Radio Association (UPRA) and the Boise, Idaho, group are working to connect with us through the Pullman, Washington/Moscow, Idaho area. South-central Washington should be on the air later this summer. We also hope to work with VADCG to get a true Gateway into their 145.65-MHz network, which uses the V-2 protocol."

Via HAMNET.

PACKET FOR THE COMMODORE 64

The manufacturer of the PKT-1, Advanced Electronics Applications (AEA) will be announcing a packet-radio adapter for the C 64 sometime this fall. The adapter will plug into the C 64 like any other program cartridge, and it will provide AX.25 protocol software, a terminal emulator program and a modem. Transmission and reception of HDLC frames will be handled by hardware. To provide enhanced HF operation, AEA may use something other than the common EXAR IC modem. How much will it cost? Around $200, but the price has not been set yet. When will it be available? This fall, probably sometime in September.

Via **WB9FLW, AEA.**

heard an ELT (outside of standard testing times). The system could then update DF information periodically or on command. Dave provides further detail6 of this idea in an article in the July issue of UPRA Connect.

Via N7BHC.

JAS-1 UPDATE

JAS-1, the amateur radio satellite built by the Japanese Amateur Satellite Corp. (JAMSAT), is complete, and should be launched in August of 1986. Along with a linear transponder operating with a 2-meter uplink and a 70-cm downlink (mode J), JAS-1 will carry a "mode-JD" digital store-and-forward transponder. This transponder will operate like an orbiting packet bulletin-board system (PBBS), and ground stations will use AX.25 TNCs and special modems to access the satellite.

With two flight-ready satellites complete (a primary spacecraft and a backup), specifications of the mode-JD transponder are becoming available. The following detail6 come from the "Satellite Update" column in Amateur Radio Action, an Australian ham magazine.

The transponder has four uplinks in the two-meter band and one downlink in the 70-cm band. The single downlink will be able to keep up with multiple uplinks because there will be frequent uplink collisions between ground stations that can not hear each other. The uplinks, on 145.85, 145.87, 145.89 and 147.91 MHz will use 1200-bit/s, Manchester coded FM. The 435.91-MHz downlink, will use 1200-bit/s, non-return to zero inverted (NRZI), phase shift keying (PSK).

While JAS-1 will probably be available to any appropriately-equipped ground station, it is likely that most stations will access the satellitethroughteleport stations. The use of central teleports has several advantages: users without satellite antennas will be able to take advantage of the satellite, time and money invested in building special modems will be reduced and contention for the satellite's uplink channels will be minimized.

With JAS-1 mode JD, the RUDAK experiment on Phase 3-C and PACSAT all scheduled to be launched in the next year or 60, the horizons of packet radio will soon be greatly expanded.

Via Amateur Radio Action.

ARRL COMMENTS ON AUTOMATIC CONTROL

The ARRL has filed comments on the FCC's proposal to extend automatic control to all amateur operations above 29.5 MHz (RX-4879). Excerpt6 from the ARRL comments follow:

"The League reaffirms its proposal that automatic control of digital communications be permitted on all amateur frequencies above 30 MHz as outlined in its Petition. In the Notice [RM-4879], the Commission has built upon the League's request, proposing that 'any amateur station may be under automatic control, except where transmitting on frequencies below 29.5 MHz.' In order to carry out its broader proposal, the Commission proposes to modify its Rule6 dealing with third-party traffic.

"It is unfortunate that the Commission has introduced the issue of third-party traffic into this proceeding. It detracts from the primary purpose expressed in the Notice of keeping the amateur service abreast of technological developments such as computer-based message systems (CBMS) and other digital technologies. The proposed third-party Rule6 also impose unnecessary and onerous restrictions on the development and operation of digital communications networks. In fact, the proposal as presented in the Notice will not allow normal operation of CBMSs, which is contrary to one of the Commission's goals as stated in the Notice. What is supposedly allowed radio amateurs by permitting automatic control is at the same time taken away by a proposed blanket prohibition from transmitting third-party traffic by automatically controlled stations as explained below."

The comments recall that the issue of third party traffic first came to importance in the late 1970s, with the growing use and abuse of repeater autopatches. The casual use of autopatches lead the FCC to issue a news release in 1978 reading, in part, "Section 97.79(d) states that 'the licensee of an Amateur station may permit any third party to participate in Amateur radio communication from his station, provided that a control operator is present and continuously monitors and supervises the radio communications to insure compliance with the rules."' The Commission in the same notice also stated that sect ion 97.79(d) clearly "prohibited autopatching and reverse autopatching through automatically controlled repeater stations and required a control operator to be on duty at all times during these operations."

From the ARRL comments:

"What the Commission was seeking to guard against then was the origination and introduction of communications into the amateur radio medium by unlicensed individuals.

"The commission misapplies this point in the [automatic control of digital communications] proposal, however, by seeking to prohibit 'third party traffic from any amateur radio station under automatic control.' A similar potential for abuse by unlicensed individuals in a digital amateur radio system exists only at the point where the third-party traffic is originated and introduced into the amateur radio medium. It is necessary to require a control operator at this stage only. This control operator will. guard against the potential abuse of amateur frequencies in a digital system just as effectively as the control operator on duty at a voice repeater station when an autopatch is accessed. To impose the additional burden of a control operator at every point along the digital system, be it at a CBMS site or at every packet 'digipeater' along the path to the message's destination is unnecessary. Such a requirement, in fact, would severely curtail the use of the developing digital technologies because the burden on control operators would be nearly impossible to shoulder. For example, messages on a packet radio repeater are received at a rate of 1200 words per minute, and retransmitted immediately at the same rate.

Gateway

Vol. 1, No. 23
July 9, 1985

The ARRL Packet-Radio Newsletter

PACKET RADIO USED IN FIGHTING BRUSH FIRES

[The following is a report filed by Wally Linstruth, **WA6JPR** from an ongoing emergency operation. It is meant to convey the initial experiences of using a new, potentially formidable technology under emergency conditions.]

Packet radio is playing an important role in aiding firefighters involved with the Wheeler Gorge forest fire that is still raging in the forests and brushlands east of Santa Barbara, CA. Bill Talanian, WlUUQ, President of the Santa Barbara Amateur Radio Club, reports that over 200 messages have been cleared from the fire area in the last few days. The traffic consists of approximately 50 percent health and welfare and SO percent official fire department **disaster-**coordination messages for relay to the northern part of the state;

The **WESTNET** digipeater system and the **W6IXU** mailbox are being used for much of the traffic. Traffic is also being cleared to NTS through Los Angeles area hams and **WESTNET.**

The versatility and simplicity of the simplex digipeater was proven when the **KA6SOX-1** Painted Cave digipeater was easily moved to La Cumbre peak for a better path into Pendola Camp, the main fire control point. A duplex repeater was also rapidly converted to a digipeater to enhance **WESTNET** coverage of the fire area.

All is not roses, however, and we will have much more to report after the conclusion of the current activity. The problems uncovered during this emergency will provide serious food for thought for the packet-radio community. Two problems that have already been identified are:

o Line-by-line entry is tedious and error prone. This is compounded by the fact that (at least here) the people involved in packet technology are not uniformly well trained in emergency traffic-handling procedures. Also, the well-trained ARES operators are not familiar with the details of TNC operation. This problem was handled by WlUUQ who received most outbound traffic at his home where the messages were reformatted using his word processor and retransmitted to their ultimate **WESTNET** destinations.

o The California State emergency agencies have been convinced that packet is superior to means of sending messages. Because of this, **state-**purchased packet gear was being used in preference to ordinary traffic-handling methods, even though it was obvious to the control operators that traditional methods were operating at higher efficiency. This situation got worse when the **W6IXU** mailbox became congested because there is not yet a “data highway” from packet into NTS.

We can draw one moral from these experiences, even before the final report has been filed: Make sure that you have a working packet-radio traffic-handling system before you “sell“ packet to government authorities.

Expect a more detailed report soon.

From **WA6JPR.**

PACKET AT THE JACKSON-HOLE HAMFEST

The 1985 Wyoming-Idaho-Montana-Utah **(WIMU)** Hamfest, which will also be the ARRL Rocky Mountain Division Convention, has scheduled several packet-radio events. The hamfest will be held in Jackson Hole, Wyoming, August **2-4.** Here is a tentative schedule of packet-radio happenings **:**

Fri. 3 P.M. The Packet Radio hospitality room swings into action.

Fri. 7 P.M. All **packeteers** go to the Bar-J Ranch for a chuck-wagon dinner and Western variety show .

Sat. 8 A.M. Packet talks begin, continuing until 10 am. The primary speaker will be Pete Eaton, **WB9FLW.**

Jackson Hole, at the doorstep of Yellowstone National Park, is a great place to take the family if you want a combination vacation and hamfest. Plan your vacation so that you get both the best scenery and the most up to date packet radio information.

From N7BHC.

EMERGENCY LOCATION WITH PACKET RADIO

Dave Pederson, **N7BHC,** suggests a potential **public-**service use for high-altitude digipeaters. He suggest that we develop an emergency locator transmitter **(ELT)** direction finding **(DF)** receiver be interconnected with a packet station. Information from the DF receiver could be relayed via packet to search and rescue. authorities. The high-altitude sites occupied by many digipeaters would be quite good for receiving ELT beacons.

A computer-controlled ELT DF system attached to a TNC could send out an emergency beacon whenever it

EASTNET AND WESTNET LINKED

It's not the "Golden Packet," but there is now a reliable mail-forwarding link between EASTNET and WESTNET. Hank Oredson, W0RLI, author of the popular MailBox software, provides the EASTNET end of the 20-meter link, and John Eigenbrode, KD6SQ, provides the WESTNET end of the path. Messages usually take a day or so to make the cross-country trip. KD6SQ, in southern California, is not connected to the northern part of WESTNET, but messages into the Los Angeles area are reliably delivered. Harold Price, NK6K, AMSAT/VITA PACSAT manager, has regularly used the transcontinental link to communicate with the PACSAT memory group in Ottawa, Ontario. HF packet stations can be found exactly on 14.103 MHz, if you use LSB and a modem center frequency of 1700 Hz.

Ed.

REPRODUCTION OF GATEWAY MATERIAL

Material may be exerpted from Gateway without prior permission, provided that the original contributor is credited and Gateway is identified as the source.

Gateway: The ARRL Packet Radio Newsletter is published by the

American Radio Relay League
225 Main Street
Newington, CT 06111

Jeffrey W. Ward, K8KA
Editor

Larry E. Price, W4RA
President

David Sumner, K1ZZ
Executive Vice President

Subscriptions are available to ARRL members and nonmembers, at the following rates:

For 25 issues of Gateway:
- In the U.S.
  - ARRL Member $6.00
  - Nonmember $9.00
- In U.S., Canada, and Mexico by First Class Mail
  - ARRL Member $11.00
  - Nonmember $14.00
- Elsewhere by Airmail
  - ARRL Member $14.00
  - Nonmember $17.OO.

Prices are subject to change without notice. Remittance must be in U.S. funds, and checks must be drawn on a bank in the U.S.

Gateway: The ARRL
Packet Radio Newsletter
American Radio Relay League
225 Main St.
Newington, CT 06111

Nonprofit Organization
U.S. Postage
PAID
Hartford, Conn.
Permit No. 2929

weather data with various meteorological sensors coordinated by a Commodore 64. The computer communicates with a TAPR TNC, and the weather data is periodically updated and transmitted. If you are interested in this project, Dennis and Philip have detailed plans, circuit diagrams and parts lists for the construction of the sensors and interface boards. To contact them, write

Philip A. Johnson
University of Minnesota
116 Church Street SE
Minneapolis, MN 55455.

Further details on this amateur weather reporting system and the software and hardware that will make it possible will be printed in a future issue of the ARRL publication QEX.

Via **KA1DYZ.**

REPORT **FROM** ONTARIO

During the fall of 1984, a group of members of the Barrie Amateur Radio CLub with a budding interest in packet radio, decided to form a packet-radio study group. The goals of the group (called HEX 9 because there were originally 9 members) were to study the material necessary for passing the Canadian digital license exam, to establish a local-area packet-radio network, to link with Ottawa and EASTNET and to promote packet radio in Canada.

The results? A local-area network has been established, and 6 AX.25 stations are on the air. They are located at Orillia, Barrie, C.F.B. Borden, Alcona Beach and King City, Ontario. VE3LSR, in Edgar, is running a dedicated digipeater, and VE3HHW, in Alcona Beach, is running a packet-radio bulletin board on a TRS-80 computer. The link to Ottawa will be available when arrangements are completed for a digipeater in Essonville.

HEX 9 has also planned a Packet Symposium, which will be held on September 21 at Georgian College in Barrie, ON. It is hoped that the speakers at the meeting will range considerably beyond the inspirational and background material is usually dealt with in the Amateur Radio press and at packet conferences. The planners of the Ontario Packet Symposium hope to cover some of the following topics: practical advice on how to enter packet radio; digipeating, theory and practice; packet radio and existing licensing requirements in Canada and the United States, including the reciprocal licensing privileges involved; possible changes in government regulations affecting packet radio; portable packet stations; gateways and protocols; packet voice; packet radio and public service; packet radio and satellites; scenarios for the future of packet radio and the bilingual and bicultural significance of amateur packet radio in Canada.

For information on attending or speaking, contact:

HEX 9 Group
PO Box 151
Orillia, ONTARIO
CANADA L3V 633.

From **VE3NBN.**

XEROX 820 AVAILABILITY

If you are looking for a source of Xerox 820 computers, consider

BNF Enterprises
119 Foster Street
P.O. Box 3357
Peabody, MA 01961-3357.

They supply "as is" boards that they have made sure are in operating condition. For a rundown on the packet-radio applications of the 820, check Gateway issue 19.

From **W2KGV.**

MISSISSIPPI DIGITAL CLUB

A group of Mississippi Amateur Radio operators interested in packet radio met late in May to form the Mississippi Amateur Radio Digital Association, MARDA. The purposes of MARDA are to represent Radio Amateurs in Mississippi who are interested in digital communications, to provide technical expertise in the promotion and advancement of amateur digital communications technology, to act as an information exchange and to provide communications for such state and local agencies as the Office of Emergency Preparedness, Civil Defense, police and fire departments. The MARDA network will include Hancock, Pearl River, Harrison, Stone, Jackson and George Counites.

Those at the meeting agreed to support a band plan that has primary activity on 145.01 MHz and alternate channels on 145.03, 145.05, 145.07 and 145.09 MHz. On the 220-MHz band, the group proposes to use 220.4, 220.72 and 220.78 MHz for "narrowband" activity, and 221.57 MHz for a 100 kHz wide highspeed packet channel.

MARDA
c/o Joe Butler, K5OS
242 Woodland Circle
Ocean Springs, MS 39564.

Via **W5CH.**

BAND OPENINGS

The weekend of the ARRL VHF contest, June 7 and 8, saw some good openings in the midwest. MailBox in Minneapolis, Minnesota, showed the following stations in its "just heard" log on Saturday, June 8: WD0CZI, KA0NCR, WA0JFS-1, WD0EMR, WA0KGU, KC0OJ, WB0HRG, WB0GGL. Several of the stations heard are from the Nebraska packet network in Fremont and Lincoln. Pat Snyder, WA0TTW, in Minneapolis, tried to establish some connections to these "DX" stations, but had no success. Pat comments: "at least a few packets made their way to the 10,000 Lakes region. Propagation such as this seems to indicate that being on one packet channel, in this case 145.01 MHz, can provide occasional DX activity." The "just heard" log on the WORLI mailbox can be a good indication of band conditions, and time-stamped packet beacons (not on network frequencies) might prove to be a valuable aid to propagation study.

Via **WA0TTW.**

Marshall, W1FJI, transferred binary f iies using TAPR TNCs and a program called PMODEM. PMODEM was written by Skip Hansen, WB6YMH, and modified for the IBM PC by Tom Cain, WB8OUE. It is written in c. If you are interested in more information, leave a message for K7PYK on one of the linked W0RLI MailBoxes; Wes'station is a major HF link in the MailBox/GateWay system.

Via **DRNET** and **WA4FIB.**

KANTRONICS RECALL

Kantronics has recalled all Packet Communicators with serial numbers below 49102. TNCs be replaced with newer units that have significantly enhanced hardware and software. The hardware changes result in reduced spurious radiations on 145.01 MHz. The early TNCs often

frequency. Communicators with serial numbers If you have one of the recalled units, ship it to:

1202 East 23rd Street
Lawrence, KS 66044.

Prom **Kantronics.**

MICHIGAN PACKET

operators in western Michigan are

Radio Association (WMPRA). Write:

WMPRA
c/o Len Todd
1819 Eloise Avenue
Muskegon, MI 49444.

Packet radio is also picking up steam in southeastern Michigan, and Eastern Packet Radio Of Michigan (EPROM), is coordinating activity in that part of the state. David Wang, N8BMA, has just established a MailBox in Detroit, and EPROM's plans call for a 220-MHz network linking eastern Another of the club's projects is "MNET" a multiuser bulletin-board system that will have both telephone and The packet network in southeastern Michigan, 147.555

July. To contact EPROM, write to

WB8TKL

Monroe, MI 48161.

Via Len Todd and **KC8OH.**

WISCONSIN CLUB

The Wisconsin Amateur Packet Radio association (WAPR) has scheduled a meeting for all interested packet-radio operators to discuss the future of packet radio in Wisconsin. The meeting will take place at the South Milwaukee Hamfest on Saturday, July 13 at 11:00 A.M. in the Oak Creek VFW Building, near Milwaukee airport. For more information, contact:

WAPR
c/o David Boede, WD9ESH
PO Box 1215
Fon Du Lac, WI 54935.

From **WD9ESH.**

NYC DIGIPEATER

There is now a digipeater in York City on the EASTNET frequency, 145.01 MHz. John Martin, WB2VTN, reports that he just got involved in packet and has placed a dedicated digipeater, WB2VTN-1, on the roof of a tall building in midtown Manhattan. The system seems to work well, with reliable connections to several major nodes EASTNET, WB2KMY, KG1O, WA2SNA, AI2Q, N2DSY.

From **WB2VTN.**

NORTH CAROLINA CONTACT

K4CAV, Digital Communications Society, predominantly an RTTY club, is getting interested in packet radio.

"fuel to the fire," contact

PDCS
K4CAV
4101 Summerglen Court
Greensboro, NC 27406.

From **K4CAV.**

ATLANTA HAMFEST

There will be a packet-radio forum and a computer forum at the Atlanta HamFestival, July 6 and 7. The HamFestival, sponsored by the Atlanta Radio Club, will be held in the Georgia World Congress Center, in Atlanta. If you want to meet some of the SOUTHNET packeteers, plan to attend.

Via **W4RH.**

VETERAN'S PACKET NET

Roger Cooper, N3RC, is looking for volunteers to help establish a packet-radio network among U.S. This HF packet network would employ some rigs that are already in and there are funds available to purchase more rigs and packet If you would like to volunteer, telephone Roger at 202-389-5237.

Via **HAMNET.**

PACKET WEATHER NET UPDATE

Although this information is a bit dated, packet operators should be interested to know that some progress has been made toward integrating amateur weather stations and amateur packet-radio stations. Philip Johnson and Dennis Moon, at the University of Minnesota School of Physics and Astronomy, demonstrated a prototype packet weather station last fall. They have been collecting

Gateway

Vol. 1, No. 22
June 25, 1985

The ARRL Packet-Radio Newsletter

FLORIDA PACKETFEST

The Florida Packetfest held June 14 and 15 has received rave reviews. The following report is from Jack, WA4FIB.

"There were about 125 packeteers in attendance, from Florida, Georgia, Mississippi, Maryland, Illinois and a few other states. Gwyn Reedy, W1BEL, did a fine job of setting up the seminars, and the Gators at the University of Florida Amateur Radio club provided a great site for the convent ion. There were seminars on the TAPR TNC 2, 9600-bit/s linking, the Gator-1 (and future Gator-2) networks, digital picture transmission (you should see the super pictures!), packet networking and the Xerox 820. There were often two seminars running concurrently, and some had to be repeated so that everyone could attend. Those who missed the gathering really missed a good, informative weekend. But fear not, there is already talk of another gathering within the next year."

Pete Eaton, WB9FLW, also attended the Packetfest, and suggests that other packet groups around the country try to hold their own regional packet convent ions. There are many hotbeds of packet activity where it would be easy to attract 100 or more interested people to a weekend of seminars and discussions.

From WB4FIB and WB9FLW.

QEX PACKET ARTICLES

There are two items in the June issue of QEX: The ARRL Experimenters' Exchange that might interest packeteers. First is a two-page article by Frank Roberts, VE3FAO, titled "Putting the Azden PCS 3000 on Packet." Frank believes that the Azden's 2-ms TR switching time makes the rig a good choice for packet radio, and he provides detailed information on how to correctly connect the necessary audio, PTT and squelch signals. The second article, by Robert Ball, WB8WGA, describes a simple, l-chip DC power modification for the TAPR TNC 1.

Ed.

PACKET IN QST. TOO

The July issue of QST will contain the first article in a two-part packet radio series by Harold Price, NK6K. The article, "What's All This Racket About Packet?" answers the questions "what is packet?" and "what can it do?" It also provides some operating tips and a list of the frequencies used by packeteers in various parts of the U.S. If your friends have been asking you what the strange noise coming from your radio is, have them read Harold's article. The second article in the series, scheduled to be printed in the August issue of QST, provides a closer look at the TNC and the connections between the radio, the computer and the TNC. Congratulations to Harold for writing a fine, informative article.

Ed.

BINARY TRANSFER PROTOCOLS

Jack Brindle, WA4FIB, author of the MacPacket/TAPRterm software mentioned in Gateway issue 21, has also been at work on a protocol and software to solve the problem of sending binary files over packet links. A complete definition of Jack's protocol will probably appear in the July issue of PSR Quarterly, but the outlines of it are presentedhere. Essentially, it is a transport (ISO RM level 4) and session (level 5) protocol with some elements of an application layer (level 7). It includes several service classes and error detection and correction at the transport lager. The use of true sessions at the session layer will allow multiple connections from one terminal to numerous other stations, True multiple-connection capability will have to wait for improved link and network layer software, but terminals, incorporating Jack's protocol will be able to perform several functions simultaneously even with the current TNC link layer. This means that you could transfer a file and hold a conversation simultaneously,

Jack is not the only person experimenting with binary-file transfer protocols. A message on CompuServe HAMNET from Mark Matthews, WA6LZO, provides a detailed look at a simple file-transfer mechanism for packet radio. Mark's protocol can transfer files of any length, using almost any file-name format. Either the station wishing to receive a file or the station wishing to transmit a file can initiate a transfer, and there is a flexible error-reporting system. Mark uses some six-character attention sequences to synchronize the file transfer, and transmits the name and length of the file in a specially-formatted header. The transfer protocol assumes that the AX.25 (TM pending) link will deliver error-free, correctly ordered data to the receiving station, but there is a "crash timer" and an acknowledgment sequence to detect link failure.

In yet a third file-transfer project, Wes Morris, K7PYK, reports that on June 16, he and Art

EASTNET HF NET

Participation in the EASTNET 75-meter SSB net has dropped to only two or three stations. To try to generate more interest, the unofficial net control station has decided to move the net to Thursday night, 9:00 P.M. EDT. The frequency remains 3.855 MHz. If you are interested in the growth and design of EASTNET, please check into the net. If you are new to packet radio and want some questions answered, turn on your HF rig and ask the assembled "experts."

Via WB2KMY.

TAPR CLOSES HF NET

Poor propagation and lack of net-control stations have caused TAPR to discontinue its 15- and 20-meter HF nets.

Via PSR.

REPRODUCTION OF GATEWAY MATERIAL

Material may be exerpted from Gateway without prior permission, provided that the original contributor is credited and Gateway is identified as the source.

Gateway: The ARRL Packet Radio Newsletter is published by the

American Radio Relay League
225 Main Street
Newington, CT 06111

Jeffrey W. Ward, K8KA
Editor

Larry E. Price, W4RA
President

David Sumner, K1ZZ
Executive Vice President

Subscriptions are available to ARRL members and nonmembers, at the following rates:

For 25 issues of Gateway:
- In the U.S.
  - ARRL Member $6 .00
  - Nonmember $9 .00
- In U.S., Canada, and Mexico by First Class Mail
  - ARRL Member $11 .00
  - Nonmember $14.00
- Elsewhere by Airmail
  - ARRL Member $14.00
  - Nonmember $17 .OO.

Prices are subject to change without notice. Remittance must be in U.S. funds, and checks must be drawn on a bank in the U.S.

Gateway: The ARRL
Packet Radio Newsletter
American Radio Relay League
225 Main St.
Newington, CT 06111

Nonprofit Organization
U.S. Postage
PAID
Hartford, Conn.
Permit No. 2929

CONNECTICUT PACKET CLUB

Connecticut now has its own packet radio club, the Southern New England Association of Packeteers (SNAP). The club will assist the development and organization of packet radio in Connecticut and strive to integrate the Connecticut segment of EASTNET into NTS. There have already been two meetings of SNAP, although the election of officers and the ratififcation of constitution and bylaws have been delayed until the July 2 meeting. The club will meet on the first Tuesday of every month, at 7:30 P.M. in the ARRL Headquarters in Newington, Connecticut. SNAP will hold a Z-meter FM net, Wednesdays at 9:30 P.M. on 147.75/.15 MHz.

From **KE3Z.**

CONNECT INDICATOR

If you are wondering what to do with the signals available from the parallel port of your TAPR or Heath TNC, the Maxtec "KON-D-KON INDICATOR" may be for you. KON-D-KON is a small (1.5- X 1.5-inch) circuit board that connects to your TNC and a speaker and sounds a short tone when someone connects to you. When KON-D-HON is connected, the "spare" LED on the TNC indicates connect status. KON-D-KON is available as a bare PC board, a complete kit or assembled and tested, from:

Maxtec Writing
3721 Spring Valley No. 111
Addison, TX 75234.

Max Adams, W5PFG, maker of KON-D-KON, reports that the gadget helped him work W4WEB near Panama City, Florida. Max was not at his terminal when W4WEB connected, but the KON-D-KON alerted him and he was able to complete the 600-mile contact.

From **W5PFG.**

NTS, PACKET AND ASSISTANT TECHNICAL COORDINATORS

In the Eastern Massachusetts section, packet radio was united with the National Traffic System (NTS) to help find volunteers for the new ARRL Assistant Technical Coordinator (ATC) appointments. Luck Hurder, KY1T, and Mort Cohan, KA1IU, used a list of ARRL members in the EMASS section to address several hundred radiograms. The radiograms were stored on the K1BC PBBS, which is dedicated to traffic handling, and NTS members with packet stations either delivered the messages or relayed them to the NTS Section net. The radiograms were delivered quickly and resulted in responses from more than 30 qualified candidates. The support of NTS by packet networks is important, and we are interested in hearing from people who are developing connections between the two communications systems.

Via **KY1T.**

BINARY FILE TRANSFERS

Many packeteers want to transfer binary files using packet radio, and this is one of the "services" that will be demanded of future packet networks. Several recent messages on CompuServe HAMNET discussed attempts to use telephone data-transfer protocols like XMODEM and KERMIT to transfer data over packet-radio links. The delays introduced by the packet channel make these timing-sensitive protocols nearly useless. The packet-radio community needs to agree on a simple but expandable protocol for binary transfers. The protocol need not detect or correct errors, since the AX.25 link will take care of these things. The protocol must provide a means for the sending station to find out when the receiving station is ready, and it must. provide some data-transparent mechanism for detecting the end of the transfer (some kind of byte counting mechanism, perhaps). If you are working on such a protocol, it is time that you made it public. Consider writing an article for one of the packet radio newsletters.

Via **HAMNET.**

220-MHZ EXPERIMENT

Norm Sternberg, W2JUP, is running a packet beacon on 221.02 MHz from midnight to 8 A.M., EDT. Ttie beacon runs 25 watts to an omnidirectional antenna in Farmingville, Long Island (grid square FN30). This experiment should provide an indication of the propagation that will be faced by the EASTNET stations planning to install 220-MHz, 9600-bit/s links. Norm would like anybody who hears the beacon to report to him at 516-698-8818, or on his BBS at 516-726-2208.

Via **HAMNET.**

MACINTOSH PACKET PREPROCESSOR

If you have a Macintosh computer and a TAPR-compatible TNC, you will be excited by the announcement of the MacPacket/TAPRterm. This "front-end processor" turns your Macintosh and TNC into a ccmplete computer communications system. It features a split-screen display, pull-down menus containing all TNC commands and parameters, file uploading and downloading, and a routing file (so that you don't have to remember the calls of every digipeater to connect to someone>. The program makes extensive use of the famous Macintosh user interface to help you make your packet station more "friendly." The split-screen display ends the problems of deciding who typed which characters. TNC parameters may be modified by "pulling down" one of the TAPRterm menus. To connect to another station, you simply select "Connect" from the command window, type the call sign of the station that you want to connect to and activate the automatic routing system. Uploading and downloading of text files and printing of received data are currently supported. Binary data transfer will be available shortly. The software is available from:

Brincomm Technology
19451 Gulf Blvd. #503
Indian Rocks Beach, FL 33535.

The $39.95 price includes all updates for one year

Via **HAMNET, WA4FIB.**

communications, He suggests that packet-radio clubs carry out their own emergency drills before trying to impress public-safety agencies with packet radio.

For those who live in the northeast, the New England Packet Radio Association (NEPRA) prints the Packetear. This is another newsletter with items of both local and general significance.
As packet radio becomes more and more popular and commercial equipment becomes widespread, clubs that are not dedicated to digital communications are getting interested in packet. This is reflected in numerous newsletters that we receive from Amateur Radio clubs throughout the country. The newsletter of the Mid-Atlantic Amateur Radio Club reports that the club will support a digipeater and a packet bulletin-board system (PBBS) in the near future. The Ramapo Mountain Amateur Radio Club (New Jersey) already supports WA2SNA, a major digipeater in EASTNET. The most recent issue of their journal contains an overview of packet in the U.S., a map of the major links in EASTNET and a report on high-speed linking efforts. On the west coast, the Boeing Employees' Amateur Radio Society (BEARS) digital communications committee is assembling a digital position at the club's station. The position will include packet radio, of course. The Palo Alto Amateur Radio Association newsletter, PAARA Graphs, is filled with packet-radio information, including a WESTNET map and some messages captured from the local PBBS.

These items are presented to show that packet radio is spreading, and that both special-interest and general-interest Amateur Radio Clubs are talking about digital communications. If your local club isn't on packet yet, consider writing an article for their newsletter. Editors are always looking for items to print. If your packet-radio club publishes a newsletter, please be sure to put Gateway on the mailing list.

From Various **Newsletters.**

CONSTRUCTION CLASS

Bob McCormick, KA1KPH, has come up with an interesting way to get members of the Hampden County Radio Association onto packet radio. Bob is going to hold a "Let's Build a TNC" class during which club members will build Heath HD-4040 TNCs and learn to use them. He hopes that this will familiarize the attendees with good construction techniques and give them some guidance during the "now that its built, what do I do with it?" phase. This sounds like something that a club might plan in the summer and execute in the fall, giving everyone something to look forward to. (Bob should also be congratulated for his two-part introduction to packet that was printed in recent issues of World Radio.)

Via **HCRA** Zero Beat.

CABINETS FOR TAPR **FROM** HEATH

If you didn't get a custom cabinet for your TAPR TNC 1, and you've still got your TNC mounted on plywood or sitting on the bench, Heath has come the rescue. A call to Heath's parts department confirmed that they have put together a package of parts that contains everything that you need to put your **TAPR TNC** 1 into the brown, low--profile cabinet that Heath's HD-4040 comes in. The parts included are the chassis, cabinet top, low-profile transformer, line cord, power switch, LEDs and hardware. Total price for the parts is just under $40, and Heath has everything in stock but the cabinet tops. Since TAPR has no more cabinets, this is probably the best deal going for TNC 1 owners who want cabinets. To order, call Heath at 616-982-3571.

From **N1CB** and Heath.

PACKET AND THE IEEE

May was a good month for packet radio in the professional electronics press. The Institute of Electrical and Electronics Engineers (IEEE) is a professional organization that prints many publications, both general and specialized. In the May issue of IEEE Spectrum, the magazine received by all IEEE members, there is an article titled "The High-tech Hobbyhorse," discussing how "personal computers and computer-based systems are turning two of the oldest hobbies in the United States -- amateur radio and model railroading -- into high-tech pastimes." Two packeteers, Harold Price, NK6K, and Terry Fox, WB4JFI, were interviewed for the article, and packet is discussed extensively under the heading 'A network node in your home?" There is also a photograph of Harold's station; a TAPR TNC and several computers show clearly. The technical IEEE Journal on Selected Areas in Communications, in a special issue devoted to communications for personal computers, carried article called "Packet Radio in the Amateur Service." This article was written by Phil Karn, KA9Q, Harold Price, NK6K, and Bob Diersing, N5AHD. Congratulations to all involved for pointing out to the professional community that Amateur Radio is still state of the art.

Ed.

ITU FASCICLES AVAILABLE

What is a "fascicle" and why is it in Gateway? Well, Webster's gives us the interesting definition "1 a: an inflorescence consisting of a compacted cyme less capitate than a glomerule" and the more relevant "2: one of the divisions of a book published in parts." In this case, we are talking about some fascicles published by the International Telephone and Telegraph Consultive Committee (CCITT) detailing their R, S, and U series Recommendations. Volume VII - Fascicle VII.1, covers telegraph transmission (R series) and telegraph services and terminal equipment (S series). Volume VII - Fascicule VII.2 is the U series Recommendations for Telegraph Switching, including telex internetworking. The two fascicles, or "fascicules" are available for 34 and 31 Swiss francs, respectively. Write to:

International Telecommunication Union
General Secretariat, Sales Service
Place des Nations
CH-1211 Geneva 20
SWITZERLAND.

Via **W4RI.**

Gateway

Vol. 1, No. 21
Jun. 11, 1985

The ARRL Packet-Radio Newsletter

GATEWAY DELAY

This issue of Gateway is the one that you expected on June 4. It was postponed until June 11 because the editor was on vacation. Gateway will now resume its schedule of an issue every two weeks; issue 22 will be printed on June 25, issue 23 on July 9, and so on. This one-week shift of the schedule also helps us with another problem: summer is a slow period for Amateur Radio, and there isn't as much news as we would like. Please send items for Gateway, either through the mail to ARRL Headquarters, via CompuServe (our user number is 75105,737), or via DRNET (you may have a DRNET member in your area>. We are especially interested in reports of Field Day packet activity.

Jeff Ward, Editor.

INTERNATIONAL PACKET NEWSLETTER

We have received a "beta test" issue of PacketWorld - - International Journal of Packet Radio. The newsletter, published by anew club called Packet Telecommunications International, will address "packet radio for hams, computer enthusiasts and businesses." PacketWorld will come out every two months, and a one-year, six-issue subscription is $10. The first issue, due out in July, will include articles on protocols, regional networks, hardware and software, mobile data terminals and bulletin boards. Packet Telecommunications International meets regularly in Santa Rosa, California. To subscribe or contribute to PacketWorld, contact:

PTI West
520 Mendocino Ave, Ste 340
Santa Rosa, CA 95401.

From PacketWorld and KA6ATN.

PSR GOES QUARTERLY

The Packet Status Register (PSR), newsletter of the Tucson Amateur Packet Radio Corp. (TAPR) will become the PSR Quarterly in July. The folks at TAPR believe that the PSR's role as a carrier of timely operational and organizational news is being filled by other publications and that what the packet-radio community needs is a technical journal carying items of lasting significance. The new PSR Quarterly, scheduled to be published every July, October, January and April, will fill this need.

Gwyn Reedy, WlBEL, will be the editor of the new publication. Judging by the quality of Gwyn's editing of the FADCA>BEACON, PSR Quarterly will be an attractive, informative publication. Remember, however, that a successful technical publication needs more than a skilled editor; it needs good technical articles. If you want to submit an article for the July issue of the PSR Quarterly, it should reach Gwyn by June 25. Send articles to:

Gwyn Reedy, WlBEL
Editor, PSR Quarterly
812 Childers Loop
Brandon, FL 33511.

Via PSB and DRNET.

NEWSLETTERS

We receive several packet-radio newsletters, some intended for local distribution and some national in scope. Without reprinting entire newsletters, we would like to present some of the highlights of those received recently.

FADCA>BEACON is the journal of the Florida Amateur Digital Communications Association (FADCA), but this large newsletter carries articles that are interesting for all packeteers. For example, the May issue of FADCA>BEACON contains several columns, including one for beginners, and features on the TAPR TNC 2, the Kantronics TNC, 9600-bit/s linking, a double-density disk controller for the Xerox 820 and the proposed TAPR network controller. There is also an index to all previous FADCA>BEACON articles. The layout and artwork in this newsletter, done by Brad and Cheryl Voss, make the FADCA>BEACON a pleasure to read. Back issues of FADCA>BEACON are available for a reasonable price. For more information on FADCA and its newsletter, write FADCA, c/o Gwyn Reedy, WlBEL, at the address in the "PSR GOES QUARTERLY" item in this issue of Gateway.

UPRA CONNECT is published by the Utah Packet Radio Association (UPRA). The May issue of the UPRA CONNECT is 11 pages long, and it provides local news and reprints from other newsletters. Dave Pedersen, N7BHC, reports that the Salt Lake City group hopes to to establish links to Boise, Idaho, Anaconda, Montana, and Grand Junction, Colorado before this winter. UPRA has prepared an information package that it will send to groups interested in helping establish these intermountain links. In another UPRA CONNECT article, Steve Peterson, KI7L, discusses the important topic of Amateur Radio emergency

have the date of the transaction available and information on the method of payment. If you call after hours, you will get the answering machine; Please leave a number where you can be reached during the day.

Please do not call about any of the following items:

- Orders for TNC-2
- New TNC-1 kit or cabinet orders
- 9600-bit/s modems
- TNC software or hardware questions

Both TNC-2 and the 9600-bit/s modem boards are in beta test now. TAPR will begin taking orders when they are available.

The TAPR phone number is (602) 746-1166.

From NK6K.

REPRODUCTION OF GATEWAY MATERIAL

Material may be exerpted from Gateway without prior permission, provided that the original contributor is credited and Gateway is identified as the source.

Gateway: The ARRL Packet Radio Newsletter is published by the

American Radio Relay League
225 Main Street
Newington, CT 06111

Jeffrey W. Ward, K8KA
Editor

Larry E. Price, W4RA
President

David Sumner, K1ZZ
Executive Vice President

Subscriptions are available to ARRL members and nonmembers, at the following rates:

For 25 issues of Gateway:
- In the U.S.
  - ARRL Member $6.00
  - Nonmember $9.00
- In U.S., Canada, and Mexico by First Class Mail
  - ARRL Member $11.00
  - Nonmember $14.00
- Elsewhere by Airmail
  - ARRL Member $14.00
  - Nonmember $17.00.

Prices are subject to change without notice. Remittance must be in U.S. funds, and checks must be drawn on a bank in the U.S.

Gateway: The ARRL
Packet Radio Newsletter
American Radio Relay League
225 Main St.
Newington, CT 06111

Nonprofit Organization
U.S. Postage
PAID
Hartford, Conn.
Permit No. 2929

SIGHTLESS HAM ON PACKET RADIO

April 25, 1985 marked an important event for the Rochester packet-radio fraternity. 1985 has seen rapid growth of packet in the Rochester area, but the highlight has to be the appearance on packet of Walt Keleher, KA2ASL. What's so unusual? Walt is blind.

Operating on voice or CW without sight presents some challenge, but think about operating on packet radio without your sight. Sending is not a big problem, since Walt can touch type, but receiving is another story. Walt has an Apple IIe computer, an Echo voice synthesizer and a specially developed terminal program. This terminal program unites the computer/terminal and the text-to-speech functions of the synthesizer or "talker." Whatever appears on the screen of the computer is "read" by the talker, in nearly perfect speech. If pronunciation errors are made, Walt can back up to get a repeat or to have the unit spell out the unknown word or phrase.

On the evening of April 25, Dave Denz, N2DWL, and Mark Winrock, both members of the Rochester packet group, went to Walt's home and set up his TNC and radio. After a short period of instruction and coaching, the big moment arrived. The equipment worked perfectly. Walt worked WB2NBU, K2YNW and W2DUC in rapid succession. While Walt was taking his time learning about packet operation, the rest of the gang was taking turns sending to him. Walt reports that we are perfect copy, but all seem to have a Swedish accent (a common trait of text-to-voice processors).

It was exciting evening, introducing Walt to packet radio and helping willpower overcome a great barrier.

From W2DUC.

CONTACTS IN THE SOUTH

At a recent packet-radio forum at the Baton Rouge, Louisiana hamfest, several amateurs from Louisiana and Mississippi expressed the need for better communications among the scattered centers of packet radio in the southern U.S. If you are interested in helping connect the western reaches of "SOUTHNET," write:

Ken Shutt, K5GUU
12433 Archery Dr.
Baton Rouge, LA 70815

or

Alan Clark, WD5IKD
2325 Milam Street
Pearl, MS 39208.

In areas where packet stations are few and far between, it is important that those who are on the air or interested in getting started can get in touch with others who've been bitten by the packet bug.

Via WD5IKD, K5GUU.

INTERMOUNTAIN REPORT

Dave Pedersen, N7BHC, of the Utah Packet. Radio Association (UPRA), recently gave a packet-radio introduction and demonstration to more than thirty people in Boise, Idaho. There are several packet stations under construction in Boise. Dave, located in Salt Lake City, Utah, reports that he has been able to make a 300-mile direct connection with Jeff Bishop, W7ID, in Boise. Unfortunately, the path is not quite solid on 2-meter FM. Dave and Jeff regularly hold Z-meter SSB skeds, and they hope to try some SSB packet (real FSK instead of AFSK on FM) soon. A digipeater link between Salt Lake City and Boise should be completed this summer.

From N7BHC.

HF PACKET

If your VHF packet network is getting a bit crowded or you just want to try out another facet of packet radio, connect your TNC to your HF rig and give HF packet a try. HF packet stations transmit at 300 bauds, using a 200-Hz shift. While some stations are using Bell 103 modems modified to transmit and receive on one set of tones (simplex), any 200-Hz shift modem can be used. If your transmit tones are the same as your receive tones, just tune in another station and you're on your way. Packet-radio employs NRZI encoding, so which sideband you use is not important. Most of the HF packet activity is on 30 and 20 meters, with 10.147 MHz and 14.102 MHz the popular frequencies. The following stations are running the WORLI Gateway software on HF packet: KI0Q, Ames, Iowa; N4CI, Conyers, Georgia; KF4JF, Hahira, Georgia; K7PYK, Scottsdale, Arizona; WA4SZK, Florence, South Carolina; WORLI, Westport, Massachusetts; KD6SQ, Cucamonga, California; KA1PN-1, Nashua, New Hampshire. Hank Oredson, WORLI, reports that a new Gateway station comes on the air each week. So, slow down your TNC and narrow your modem to try HF packet.

Via WORLI.

TAPB "OLD BUSINESS DAYS"

From Monday, May 20, through Friday, May 31, 9:00 AM until 5:30 PM PST, TAPR will double its office staff and disconnect the telephone recorder. During this time they will have a human answering the phone, and they will be returning the backlog of calls that has built up.

What exactly is "Old Business Days"? It's a marathon session during which TAPR hopes to clear out the last loose ends of the TNC-1 product line. Call if you have any outstanding business with TAPR regarding any of the following items:

TNC-1 Kits
TNC-1 cabinet kits.
Beta Upgrade kits.
Outstanding spare parts orders.

Also take this opportunity to order spare or replacement parts for TNC-1.

If you are calling about a backlog order, please

station setup, north and south: KA6SOX, W1UUQ, WA6CFM, WD6FPY, WB6DAO, W6IXU, N6BGW, N6CXB, NK6K, AJ6T, WB6UCK, N6ZH, NG6P, K6QIF, KB6JM, WB6HHV, W6AMT. There were 6 packet stations directly involved with traffic handling and 10 digipeaters involved in the network.

From NK6K.

IBM PC PACKET ADAPTER

Jack Botner, VE3LNY, whose article "A Packet Radio Adapter for the IBM PC" appeared in the January 1985 issue of QEX, has received many inquiries concerning software for his 8273 HDLC adapter. He reports that an implementation of the VADCG protocol is available from the Hamilton Area Packet Network (HAPN). The protocol software is written in assembler, and the support programs are written in C. To receive the software, send $15, a diskette and a disk mailer to:

Jack Botner, VE3LNY
35 Wynford Hights Circle #1708
Don Mills, Ontario
CANADA M3C 1L1

Software supporting the AX.25 protocol is not yet available for Jack's adapter. HAPN would like to hear from anyone interested in developing AX.25 software for the interface.

From **VE3LNY.**

COLLECTION OF PACKET PAPERS

If you have been reading any of the books recommended in Gateway, you have probably noticed that a few "classic" papers on packet switching are referred to over and over. Tutorial Principles of Communication and Networking Protocols, edited by Simon S. Lam is a comprehensive collection that includes many of these essential papers on digital communications. The papers cover link protocols, multiple access methods, local-area networks, resource allocation in networks, network and internet services, and protocol verification. The collection is published by The Institute of Electrical and Electronics Engineers (IEEE), and it is ISBN 0-8186-0582-o.

Via KA9Q.

PACSAT FUNDING

As reported in Gateway 15, the Radio Amateur Satellite Corporation (AMSAT) and The Volunteers in Technical Assistance (VITA), have completed engineering specifications for PACSAT, a proposed store-and-forward, packet-radio satellite. In the two months since the design and strategy meeting, no sources of the necessary funding have been identified. Recently, VITA was granted several thousand dollars to retain a professional fund raiser for the PACSAT project. While this should help, it is still important that amateurs assist by identifying potential sources of major contributions. Also, as in past amateur-satellite projects, individual and club donations to the PACSAT fund will be necessary. If you can help, contact:

VITA
1815 N.Lynn
Arlington, VA 22209
Attn: Gary Garriott, PACSAT Project Manager

Ed.

JAS-1 PACKET INTERFACE

The April issue of the Japanese magazine CQ Ham Radio has an article by JA1ANG, based on material from Miki Nakayama, JRlSWB, describing the user interface to the packet mailbox that will fly on the JAS-1 satellite. It shows a sample message from W3IWI to JRlSWB as it would be received at JRlSWB and then a reply. The dialog shown in the magazine is reconstructed below:

```
(Legend in left-hand column:
$: JAS-1 ==> User
%: User  ==> JAS-1
&: TNC <==> User)

& cmd:CONNECT JAS-1
& *** CONNECTED to JAS-1

$ Hello JRlSWB de JAS-1
$ Your last access was on 85/01/11 at 1922 UTC

$ NO. DATE   FROM TO      SUBJECT        LINES
$ 15 01/28 W3IWI JRlSWB AX.25 SOFTWARE 3

$ Command ?

% READ 15

$ Posted: Mon 85/01/28 From:   W3IWI
$ To:      JR1SWB
$ Subj:    AX.25 SOFTWARE
$ I have the disks ready to mail to you and the
$ packet is all sealed. $    Tom

$ COMMAND ?

% WRITE W3IWI

$ Subject:

% Re: AX.25 SOFT.
$ Text:

% Tom, I'm looking forward to receiving the
% packet.
%    73's Miki
% .

$ Message from JRlSWB to W3IWI saved as #18
$ COMMAND?

% BYE

& *** DISCONNECTED
```

This protocol is a combination of features of the WORLI MailBox (JR1SWB has a Xerox 820 and is running the MailBox) and of TeleMail (the message in the example was sent via TeleMail).

JAS-1 is scheduled for launch in 1986. Uplinks will be on 2 meters (145.85, .87, .89, .91 MHz) with a single downlink on 70 cm (435.91 MHz) using 1200 bauds, phase-shift keyed (PSK).

From **W3IWI.**

Gateway

Vol. 1, No. 20
May 21, 1985

The ARRL Packet-Radio Newsletter

CALIFORNIA PACKETEERS PARTICIPATE IN "QUAKE '85"

A simulated earthquake occurred on April 16 in southern California. The simulated quake was 8.2 on the Richter Scale. A quake of this size would cause a great deal of destruction and would most likely result in the immediate loss of most communications over a large area for several hours. As is the case for most large-scale exercises of this type, amateur radio played a part. This year, for the first time, packet radio made major contributions.

The goal was an ambitious one: to move traffic from the State Off ice of Emergency Services (OES) communications trailer in Los Alamitos to the OES off ice in Sacramento, 400 miles to the north, using VHF. This required the use of seven digipeater6. We learned several things during the exercise: First, it is possible to move a large amount of traffic that distance through that many digipeaters, and second, it wasn't possible to do it in the way we had originally intended.

We also had several unplanned events preceding and during the exercise which added to the simulated emergencies. First, California is blessed with a geography that provides 4000 to 8000 foot mountains and over-water paths. The length of most of the paths used in the network (called WESTNET) is 90 miles, with one path of 120 miles and another close to 200 miles. The longer paths require over-water ducts, which are in place for much of the year, Two days before the exercise, a weather pattern went through that destroyed the ducts, which didn't return for several days.

Also, three digipeaters failed the day before the exercise. This was the largest network failure experienced before or since. The systems were repaired in hours, and portable systems were readied. During the exercise, one system was driven to a mountain top to supply backup for the lost duct, and another system was "car mobile" at the OES site, ready to be driven or helicoptered should the need arise.

In addition to the amateur-owned equipment in place, the OES had previously purchased several TNCs; one was installed in their communications van in Los Alamitos and another in their office in Sacramento. A third was at an intermediate digipeater site in northern California. To make a long story short, soon after the exercise began, we established communications with Sacramento. Unfortunately, it was lost soon after, because of a flaky path at the far northern end. We didn't find this out until much later, of course. We interspersed an hour of attempted reconnects up north with passing traffic around the southern California area - San Diego, Santa Barbara, Glendale and others. This activity was very successful. After it became apparent that we couldn't stay connected to Sacramento, the goal became to get the traffic as close to there as possible. We sent several CQ messages to San Francisco, which was outside of the simulated affected area, and attracted the attention of a ham with file store capabilities. We transferred two hours of accumulated traffic in short order to San Francisco, through four digipeater hops. The San Francisco station then transferred the traffic to the Sacramento station though two hops, again with little diff iculty. Later in the afternoon as the paths improved, we were able to transfer traffic directly to Sacramento.

We learned these lessons during the test: First, two teams are needed at the packet position in the communications trailer. One team establishes and maintains the link, the other enters the traffic on a separate computer system. When a link is available, the data can be moved quickly by moving a disk from the data-entry computer to the computer with the TNC attached. During the "quake," even though two computers were available, message traffic piled up in the "IN" basket while we were trying to establish a path. Once contact was made with the San Francisco station, it took half an hour to clear the backlog. It would have taken less than five minutes had the data already been entered on disk.

We also learned that it would have been better if we had planned on an intermediate file relay station part way up the link in the first place. The cumulative effects of dropped packets increases with number of digipeaters until the probability of a packet making it all the way to the end and the ACK getting all the way back becomes small. In our case, four hops was 90% reliable (one packet not ACKed in 10), but seven hops was less that 10% reliable. Had we sent all traffic with to the intermediate station in San Francisco, we could have passed the traffic as it came in, instead of building up a large backlog.

Finally, we learned that it was possible to pass traffic a long distance through a large number of digipeaters, provided that we did it the right way. It was possible to maintain a clear channel; people with non-simulated-emergency traffic stood by or went elsewhere. We also learned that as with other types of emergency exercise, success depends on planning and on the hard work of many amateurs.

At the risk of leaving someone out, here is a list of the packet participants in "Quake 85" -- keyboarders, mountain-toppers, HQ and other off ice

DC SUPPLY FOR **TAPR** TNC

Gary Field, **WAlGRC,** is selling a converter that powers the TAPR TNC **1** from an unregulated source of **12** to **14 volts** dc. The **3-inch** by **4.5-inch** converter board requires no **modif icat** ions to the TNC. Gary's product announcement states that **"a** typical TAPR TNC draws approximately **0.75** amps when connected to 13.6-V dc." Gary is selling complete kits for $20.

If you are interested in a dc supply for your portable or emergency packet station or for your digipeater, write:

Gary Field, WAlGRC
5 Pluff Ave.
N. Reading, MA. **01864.**

Packet Radio Association has a that for the TAPR TNC. Their correct address is in Gateway Issue

From WAlGRC.

REPRODUCTION OF **GATEWAY MATERIAL**

Material may be exerpted from Gateway without prior permission, provided that the original contributor **is** credited and Gateway is identified **as** the source,

Gateway: The ARRL Packet Radio Newsletter is published **by the**

American Radio Relay League
225 Main Street
Newington, CT 06111

Jeffrey W. Ward, **K8KA**
Editor

Larry **E.** Price, **W4RA**
President

David Sumner, **K1ZZ**
Executive Vice President

Subscriptions are available to ARRL members and nonmembers, at the following rates:

For 25 issues of Gateway:
- In the U.S.
  - ARRL Member $6 .OO
  - Nonmember $9.00
- In U.S., Canada, and Mexico by First Class Mail
  - ARRL Member $11 .OO
  - Nonmember $14.00
- Elsewhere by Airmail
  - ARRL Member $14.00
  - Nonmember $17 .OO .

Prices are subject to change without notice. Remittance must be in U.S. funds, and checks must be drawn on a bank in the U.S.

Gateway: The ARRL
Packet Radio Newsletter
American Radio Relay League
225 Main St.
Newington, CT **06111**

Nonprofit Organization
U.S. Postage
PAID
Hartford, **Conn.**
Permit No. 2929

XEROX 820 ROUNDUP

The Xerox 820 is a single-board Z80-based microcomputer with 64 kbytes of RAM, two serial I/O ports, a parallel I/O port and a disk controller. In 1984, Xerox made these computers available at surplus prices, and packet-radio experimenters throughout the U.S. bought them. With hundreds of Xerox 820s in the hands of packeteers, it was not long before packet-radio applications software was available. The following is a list Xerox 820 software and hardware produced "by and for packeteers."

The MailBox and Gateway programs written by Hank Oredson, WØRLI, work with a Xerox 820, 8-inch disk drive(s) and one or two TAPR TNCs. The MailBox is an excellent packet-radio bulletin-board system, with Hank's message forwarding routines, file uploading and downloading and extensive user logging. The MailBox can be connected to two TNCs to provide a message-storage node for two networks. If you use 2 TNCs, the Gateway program allows a station connected to one of the TNCs to initiate connections using the other TNC. The Gateway is most often used to link VHF networks to HF stations but can also be used in areas where there is activity on two VHF frequencies. To receive the latest version of the MailBox and Gateway, send an 8-inch disk, a disk mailer and return postage to:

Hank Oredson, WØRLI
19 North Hill Road
Westford, MA 01886.

The '820 has enough parallel and serial I/O ports to be used as a TNC, if you have the necessary hardware and software. There are two hardware modifications used to make the Xerox handle the HDLC protocol that is the basis of AX.25. The "FAD board" is an HDLC board that replaces one of the parallel I/O chips on the Xerox with a 28530 serial communications controller (SCC). This board, designed by Howard Goldstein, N2WX, was first described in the newsletter of the Florida Amateur Digital Communications Association (FADCA). Since then, the Tucson Amateur Packet Radio Corp. (TAPR) has made bare PC boards for the FAD board available. The original run of these boards has run out, but more should become available this summer. The Z8530 chip is available (in limited quantity) from FADCA:

FADCA
812 Childers Loop
Brandon, FL 33511.

The second approach to HDLC is to use a serial I/O (SIO) chip (already on the Xerox 820) and an external state machine. The state machine recovers clock information and converts data to and from the NRZI format used by AX.25. The state-machine schematic and listings of the state-machine PROM are available from Gateway for an s.a.s.e.

When you have selected hardware for your Xerox 820, there are several software choices. FADpad is a TNC program, written by N2WX, that supports a single AX.25 link at a sustained data rate up to 9600 bit/s. The software is the same as that in the TAPR TNC 2, and the user interface is compatible with most TAPR commands and procedures. To use FADpad, you must have a FAD board. FADpad is available as a pair of customized 2764 EPROMs from:

Ted Huf, K4NT.A
1829 NW Pinetree Way
Stuart, FL 33494.

A CP/M .COM file for FADpad is available if you send an 8-inch disk and sufficient return postage to:

Howard Goldstein, N2WX
POB 905
Melbourne, FL 32901.

Phil Karn, KA9Q, has written a C program that can support multiple simultaneous AX.25 connections on a single Xerox 820. This program runs on the 820 under CP/M with either the FAD board or the SIO and state machine. Send an 8-inch disk and return postage to:

Phil Karn, KA9Q
25B Hillcrest Rd.
Warren, NJ 07060.

The '820 can also be used -- without disk drives, monitor or keyboard -- as a digipeater. Jon Bloom, KE3Z, has developed multiport digipeater software for the Xerox. The software can be used to link networks different frequencies, or it can be used as a simple single-port digipeater. It uses the SIO and two state machines for HDLC output. To receive this software, send a disk to Gateway.

Where do you get a Xerox 820 if you want one? Since the computers are no longer produced, supplies are erratic, but persistent calling to the Xerox outlet will usually turn some up. Their phone number is 214-960-3367.

If you want more information about the Xerox 820, you may be interested in Micro Cornucopia magazine. "Micro C" is dedicated to single-board computers and has a Xerox column.

Micro Cornucopia
P.O. Box 223
Bend, OR 97709

Remember, the ARRL and Gateway in no way endorse the above products. The people making this software and hardware available are amateurs, not commercial vendors; send a disk if you want one in return; include sufficient postage; be patient.

Ed.

APPLE PBBS NOT AVAILABLE

Lynn Taylor, WB6UUT, may not be able to accept any more requests for the packet bulletin-board software discussed in Gateway Issue 6 and elsewhere. If you want a copy of the software, please contact Lynn before you send any disks. If you are already waiting for return disks, they will be sent soon.

From WB6UUT.

HAL OFFERS TNC

HAL Communications Corp., a mainstay of amateur and commercial RTTY and AMTOR, has announced that it will be marketing a packet-radio controller, the RPC-1000. The gentleman at the HAL booth told me that the RPC-1000 was "essentially a TAPR TNC." The RPC-1000 is targeted primarily at nonamateur users and will sell for $995.

Another interesting item that HAL had on display was the WX-1000 "Weather Box." This device gets input from a National Weather Service Wire Hookup, and provides the following outputs: local console, printer, phone-line modem, radio modem and alarm switches. The WX-1000 might make an interesting addition to an amateur repeater or packet-radio network, since it provides up-to-the-minute weather reports and forecasts.

Ed.

TEXAS EMERGENCY EXERCISE

On April 19, eight members of the Texas Packet Radio Society participated in the Civilian Military Contingency Hospital System exercise. This exercise involved flying simulated casulties into three area airports, and transporting patients by ground ambulance, bus, or helicopter to one of seven hospitals in Tarrent county and eleven in Dallas county.

Six amateur packet-radio stations and several voice-only amateur stations participated. Five of the packet-radio stations were at sites that had never used packet radio before. All three airports, two hospitals and the city of Dallas emergency operations center (EOC) were equipped for packet radio. Each location had file storage, off-line editing and a printer. The plan was to compose a standardized message off line and send it via packet to the Dallas EOC. The EOC station would then merge the lists received from the field stations and create a sorted, master list. This list would be sent, in connected mode, to the five field stations.

The operation, while not flawless, went very well. The Parkland Hospital (Dallas County Hospital) received 24 kbytes of text in nine files. Each of these was printed and given to the hospital administration. The Garland Memorial Hospital and Carswell AFB locations also gave printouts to the officials present.

David Cheek, WA5MWD reports: "We learned several lessons. First, text entry can be painfully slow. Special attention needs to be paid to this. item. Second, different message formats can destroy any merging and sorting efforts; all stations must use the same format. Third, bring back up copies of essential software, and check computer to TNC handshaking before it is needed. Fourth, a reliable formofadcast must be developed to distribute information that is essential to all participating stations. Even with these problems, however) use of packet radio shifted emphasis from getting the messages across the radio links to getting them correctly into the links. The speed of packet radio allowed us to concentrate on tasks that we may not have had time for before."

Via WA5MWD.

NEW PBBS IN ALASKA

During a recent trip to Alaska, Tom Clark, W3IWI helped install "the farthest-north packet-radio bulletin-board system" -- KL7GNG in Cleary Summit, Alaska. The PBBS, using a Xerox 820 and the WØRLI MailBox software, serves Fairbanks on 145.01 MHz. There are now only 3 users, but within a month or two there will be several more. Demonstrating the emergency and public service uses of packet radio and the PBBS to the Fairbanks ARC generated considerable interest, perhaps because of the high number of computers per capita in Fairbanks (they have to have something to do on those long, cold winter nights!). Tom Walyer, KL7GNG, indicates that he has plans to try to establish packet links between Fairbanks and Anchorage in the near future.

Via W3IWI.

MID-ATLANTIC PACKET RADIO COUNCIL

The Mid-Atlantic Packet Radio Council (MAPRC) is a confederation of packeteers from southern New Jersey, eastern Pennsylvania, Delaware, Maryland and northern Virginia that is the network coordinating agent (NCA) for an area roughly 100 miles in radius around Elkton, Maryland. The group meets every month or so at a mutually inconvenient rest stop on I-95. MAPRC coordinates 5 major packet-radio bulletin-board systems (PBBSs) and gateway stations (W3IWI, WB2MNF, AK3P, KS3Q and WB4APR-5) and the major digipeaters in its area (WB4JFI-5, W3VD-5, WB4APR-5, WB4APR-6, K3DSM-5, WA3KXG). MAPRC plans to expand the mid-Atlantic portion of EASTNET with high-speed VHF and UHF links in the near future.

To reach MAPRC, write:

Tom Clark, W3IWI
6388 Guilford Rd.
Clarksville, MD 21029

From W3IWI.

UNIX GATEWAY ON EASTNET

Jim Kutsch, KY2D, in Little Silver, New Jersey, is now operating a gateway between the EASTNET amateur packet-radio network and "Usenet-netnews." Usenet is a worldwide collection of computers sharing news articles. These articles are called the "netnews." There are over 12,000 sites on Usenet, in the U.S., Hawaii, Korea, Japan and Europe.

The KY2D-2 gateway, a 68000-based computer running UNIX, operates on the EASTNET network frequency, 145.01 MHz, It allows EASTNET stations to read netnews articles in the "net.ham-radio" newsgroup, and provides a local newsgroup for messages of interest only to users of the KY2D system.

To use the gateway, connect to KY2D-2 and wait for the "login:" prompt, then type "guest". After you log in, you will receive some preliminary instructions. For more information on Usenet, see "Usenet: A Bulletin Board for UNIX Users," by Sandra L. Emerson, Byte, October, 1983.

Via DR NET.

Gateway

Vol. 1, No. 19
May 7, 1985

The ARRL Packet-Radio Newsletter

DAYTON

If last year's Dayton Hamvention was where packet radio "came out of the basement," at this year's Hamvention packet "came into the store." Commercial vendors Heath, **Kantronics,** GLB, HAL, AEA and Richcraft were all displaying **TNCs.** Kantronics, Heath and TAPR each had prominent packet-radio advertisements in the **Hamvent** ion program. Heath's supply of **TNCs** lasted only until Saturday morning , and dealers reported good sales of the other **TNCs.** Friday's packet forum was well attended, with between 100 and 200 people listening to three hours of presentations. A later forum on digital RF techniques was also popular. If the level of interest shown by amateurs at Dayton was a valid predictor, there will be a lot of new **packeteers** this summer.

Ed.

TAPR TNC 2, TINY TNC

After months of secret planning, designing and testing, Tucson Amateur Packet Radio Corp. **(TAPR)** has unveiled a new TNC, the TNC 2. TAPR, a nonprofit research and deve lopment organ izat ion' designed the TNC 2 to "prove new concepts and to raise money to fund other development projects."

The TNC 2, dubbed "tiny **TiNC,"** measures about 6 inches by 9 inches and is housed in a low-profile cabinet. A CMOS **Z80** CPU and other CMOS **ICs** keep power consumption below 260 **mA** at 12-V dc. The TNC 2, unlike TNC 1, does not use ac power; it requires a dc supply between 10 and 15 V. Packet **(HDLC)** I/O and clock recovery are performed by a **Z80** SIO chip and a state machine. This hardware permits full-duplex packet operation. Like the modem in the TNC 1, TNC 2 modem uses the EXAR chip set with a switched-capacitor input filter. There is a modem-disconnect header for connecting external modems. The TNC 2 comes with 16 kbytes of EPROM and 8 kbytes of RAM and can be expanded to 56 kbytes total memory. In place of the nonvolatile RAM found on TNC 1, the TNC 2 uses a lithium-battery backed-up RAM for storage of **user-setable** parameters.

The software for TNC 2 was written in **Z80** assembly language. It provides the same user commands as are found on TNC **1,** with a few additions and deletions. The support of VADCG protocols is gone, and in its place comes the ability to select either AX.25 Version 1 or Version 2. The software also keeps a list of stations heard, will send a message to connecting stations and can display the digipeater paths for monitored frames, to describe only a few of the enhancements.

The best feature of TNC 2 may be its price -- less than $200.

So that TAPR isn't deluged by phone calls, a few of the questions most frequently asked by those who saw the TNC 2 at Dayton will be answered here:

What about TNC l? The TNC 1 is no longer produced by or available from TAPR, but TAPR software support will continue. Heath, AEA, HAL and Kantronics are selling **TNCs** based on the TNC 1, so the TNC 1 is not "outmoded."

What is on TNC 1 that is not on TNC **2?** The TNC 2 has only one set of stored operating parameters, it has no ac power supply, and it has no parallel status port. **(To** replace the parallel port, TNC 2 has connect-status and buffer-status **LEDs.)**

When can I order a TNC 22 TAPR is not sure exactly when the TNC 2 will be available. There are now some units being tested, but it will probably be late summer before the first few kits are sold. Even after this, supply will probably be lower than demand.

Can I get on a waiting list? No. Wait for an announcement, or buy one of the commercial **TNCs.**

Congratulations to those responsible for both the TNC 1 and the TNC 2. While it is impossible to name all of the people involved in the TNC 2 project, it is important to note that the **"TAPR"** team includes members of several organizations, including FADCA, SLAPR, RMPRA and CAPRA. **TAPR's** efforts have been responsible for a lot of growth in packet radio, and the TNC 2 indicates that TAPR will be advancing packet radio whenever it can.

Ed.

HEATH TNC ADD-ONS

Attached to the several Heath **HD-4040 TNCs** on display at Heath's booth at Dayton were small cases with **LEDs** monitoring TNC status. These status indicators decode the signals available at the HD-4040 parallel output, and display buffer status, connect status and other TNC vital signs. The status indicator should be available by this fall and will make a nice addition to any TNC that uses TAPR hardware. Another TNC add-on that may become available from Heath is a tuning indicator for HF and OSCAR operation. This interest in packet peripherals displays Heath's hopes that packet will draw some of its computer customers toward amateur radio.

Ed.

1985, and will be considered **for** publication in the premiere edition of The **AMSAT** Technical Journal, to be published in December of 1985.

Via Amateur Satellite Report.

BOOK REVIEW

"Computer Networks" by Andrew S. Tanenbaum, ISBN O-13-165183-8. This is a very complete but readable reference on the entire subject, ranging from data transmission at Layer 1; through channel access techniques for satellite and packet radio broadcast channels; up through link, network and transport protocols (with a balanced and insightful coverage of the virtual-circuit and datagram protocols). Later chapters deal with presentat ion and application layers, with encryption and authentication given fairly good treatment. Tanenbaum is a good writer, able to explain things well and with a relaxed, often humorous style.

From KA9Q.

REPRODUCTION OF GATEWAY MATERIAL

Material may be exerpted from Gateway without prior permission, provided that the original contributor is credited and Gateway is identified as the source.

Gateway: The ARRL Packet Radio Newsletter is published by the

American Radio Relay League
225 Main Street
Newington, CT 06111

Jeffrey W. Ward, K8KA
Editor

Larry E. Price, W4RA
President

David Sumner, K1ZZ
Executive Vice President

Subscript ions are available to ARRL members and nonmembers , at the fo1 lowing rates:

For 25 issues of Gateway:
In the U.S.
ARRL Member $6.00
Nonmember $9 .OO
In U.S., Canada, and Mexico by First Class Mail
ARRL Member $11.00
Nonmember $14 .OO
Elsewhere by Airmail
ARRL Member $14.00
Nonmember $17.00.

Prices are subject to change without notice. Remittance must be in U.S. funds, and checks must be drawn on a bank in the U.S.

Gateway: The ARRL
Packet Radio Newsletter
American Radio Relay League
225 Main St.
Newingtoa, CT 06111

Nonprofit Organization
U.S. Postage
PAID
Hartford, Conn.
Permit No. 2929

amateur packet radio, with workshops covering topics as basic as how to hook up a packet-radio station and as advanced as Level-3 protocols. Other clinics planned include Packet Operating Procedures, New Equipment Demonstrations, Xerox-820 Operat ion and Modif icat ion, Commodore Computer Interfacing and TNC Theory and Interfacing.

For further information on the SOUTHNET Packet Conference, contact:

FADCA
812 Chliders Loop
Brandon, FL 33511

or

GARC
c/o Joe DiPietro, WA4NWO
5010-135 N. Waldo Road
Gainesville, FL 32601.

From WlBEL, WA4NWO.

OHIO GATEWAY

Steve Peterfreund, KAlPN, is running a WØRLI HF Gateway in Hudson, Ohio. Local users can connect on 145.01 MHz, and HF packet operators should look for KAlPN on 40, 30 or 20 meters.

via KAlPN.

PACKET IN NEBRASKA

Mike Nickolaus, NFØN, reports that there will be a packet-radio seminar at the Nebraska Mini-Convention, on May 4th. Mike also sends the following report on packet radio in Nebraska:

"Most of the packet activity in Nebraska is in the eastern half of the state. However, digipeaters are being placed in several new locations in the western part of the state, and a link from the eastern border to the western border is near completion. All Nebraska digipeaters are operating on 145.01 MHz. The Tekamah digipeater is the link that ties Nebraska to Iowa, and the KCØOJ MailBox, in Battle Creek, Iowa, has inputs on both 145.01 MHz and 147.555 MHz (the frequency used by most Iowa stations)."

From NFØN.

HIGH-SPEED LINKS

There will soon be a test link between downtown Chicago, Illinois and La Port, Indiana using the 9600-bit/s, 220-MHz modems designed by Steve Goode, K9NG. The link will initially use a Xerox 820 board with multiport digipeater software. This is also what the WESTNET and EASTNET experimenters intend to do to enhance their current 2-meter digipeater paths. Groups in Florida and Iowa are also going to be using the 9600-bit/s radios with their own Xerox 820 software. Each group experimenting with the 9600-bit/s links should keep the other groups informed.

From K9NG.

NETWORK-LEVEL PROJECTS

The third meeting of the CAPRA layer-3 networking committee was held on April 18th. Discussion centered on methods of implementing the datagram protocols TCP and IP over the CAPRA test link between Indiana and Chicago. Installation of the 220-Mhz radios and antennas for the link is proceeding on schedule, and should be complete sometime within the next two months. Source code (in the C Language) for TCP and IP will be acquired within the next two weeks, and the committee will examine this code to see what changes have to be made to make it fit for amatuer radio use. The committee is also examining the 68000 microprocessor as a network-control processor (as are a number of other groups).

From K9NG.

FAD BOARD PARTS

The Florida Amateur Digital Communications Association (FADCA) has made a purchase of Z8530A SCC chips for use with the FAD board on the Xerox 820. These ICs are available for $20 postpaid; this is a remarkably good price on a hard-to-find part. If you need an Z8530A, write:

FADCA
812 Childers Loop
Brandon, FL 33511.

The FAD board replaces the parallel I/O chip on a Xerox 820 and is capable of handling 2 synchronous packet channels. Software for the 28530 is being developed by several groups, including FADCA and AMRAD.

via WlBEL.

PROJECTS UNDER WAY

Several people have commented to me that they do not know what technical projects are going on and who is working on them. This lack of communication has not yet caused any large duplication of effort or "reinvention of the whee 1," but it is surely slowing down certain projects. Although Gateway is not a technical newsletter, I will try to provide more news of technical interest. Please contact Gateway if you are working on a project that might be interesting to others. Don't wait until you are finished to tell the world what you are doing; saying that you are investigating a problem is not a commitment to come through with a completed product.

Ed.

AMSAT CALL FOR PAPERS

AMSAT, The Radio Amateur Satellite Corp., Inc. has issued a call for professional papers reporting original work and/or signif icant findings in the field of low-cost satellite engineering, space communications, space sciences and related soc ial value issues. Suggested topics that are of interest to packet-radio experimenters include digital voice-encoding techniques, packet-radio techniques with emphasis on higher-level networks and teleports. Papers are due before August 1,

traffic and general damage estimates were handled on the Z-meter ARES voice nets. Experiments were conducted using packet radio to replace all outgoing police calls (all outgoing phone 1 ines from the police station were jammed) and to use packet radio for health and welfare traffic between an emergency relocation center and the police station. Attempting to replace voice communications (the downed' phone lines) with packet did not prove successful. Packet proved to be usable for the health and welfare traffic between the relocation center and the police stat ion. All in all I think the test went very well, especially considering that the packet stations were assembled the night before the drill. Two portable packet stations were provided for the links at the relocation center and the police station. A mobile digipeater was also provided. I think the main points brought out by the drill are that a standard for TNC connectors would be helpful for throwing together emergency packet stations and that a standard packet health-and-welfare form should be agreed upon. Also, software to allow easy entry of messages in this form would be useful. Such software could be used to link emergency relocation centers, the morgue and a computer center. Lists of people at each center and the morgue could be kept at the central computer. Anyone looking for relatives would then ask the local relocation center to check the data base and see if they are at another center. I realize none of these observations are new but I would like to give encouragement anyone th ink ing of writing this kind of software."

[In the March issue of the Packet Status Register, (newsletter of the **Tucson Amateur Packet** Radio Corp.) Jay Nugent, **WB8TKL,** sets forth a proposal for a standard **"PACGram"** -- a NTS radiogram sent via packet. Jay has written some software that facilitates entering and printing **PACgrams. - Ed.]**

From **K9NG.**

TRENTON PACKET FORUM

The day-long packet-radio form at the Trenton Computerfest (April 20th) was attended by **packet-**radio enthusiasts from the length of **EASTNET.** Encouraging reports were heard from all parts of the network.

Interest in **220-MHz** linking was high, and some preliminary agreements were made, In general, those interested in high-speed linking agreed to use the **K9NG** FSK modems and the **KE3Z** multiport digipeater software. Ed Picchetti, **K3RLI,** is already running a **1200-bit/s, 220-MHz** link between Wilkes-Barre, Pennsylvania and **KB3ZW,** in Honesda **le,** PA. Packets to be transmitted between these two stations are automatically switched from 2 meters to 220 MHz, without user intervention. The **KE3Z** multiport digipeater software does much the same thing, with more versatile routing. Further discussions of **EASTNET** high-speed linking will take place on the informal **EASTNET** discussion net, Sundays at 8 P.M. on 3.855 MHz.

The same network crowding that is making the **220-**MHz projects important made operating courtesy a hot discussion topic. The availability of several commercial **TNCs** will certainly result in an increase in the number of novice packet-radio operators on the air. With increasing network population, it is important that we all follow a few simple operating rules:

1) Send CQ packets only when you are actively seeking to contact someone, not whenever your station is on the air.

2) Send QST packets only if you have information of immediate interest to everyone on the network.

3) If you must send QST or CQ packets, send them only through one digipeater and send them infrequently.

4) Send packets of reasonable length. Packets of 40 or 80 characters are much less likely to be in collisions than longer packets.

5) Monitor the network before you connect to a bulletin-board system. Remember that a PBBS sending a long newsletter or message may monopolize a frequency; perhaps you can wait and connect to the PBBS when other stations are not using the network.

6) Always move off of the main network frequency (145.01 MHz for **EASTNET)** if you are in a direct (no repeater) connection,

Those at the Trenton forum, and experienced packet operators everywhere, encourage everyone to follow these rules and to remember that every packet that sent affects every station on the network.

Thanks to Jon Pearce, **WB2MNF,** and Harold Winard, **KB2M,** for organizing the events at the Trenton Computerfest.

Ed.

TEXAS PACKET CLUB

The Texas Packet Radio Society **(TPRS)** is taking shape in the **Dallas-Ft.** Worth area. TPRS has voice net Wednesday nights at **8** P.M. on the **146.01/.61** MHz repeater. David Cheek, **WA5MWD,** is running a WORLI **MailBox** on 145.01 MHz, and is looking for other **MailBox** operators in Texas, Arkansas or Oklahoma that are interested in message forwarding. Traffic handlers in the area are looking for a way to use the **MailBox** for message handling. TPRS expects to have a **wide-**coverage digipeater operation on 145.01 MHz by the end of April, and they want to hear from anyone interested in constructing regional network. For further information, contact:

Texas Packet Radio Society
P.O. Box 835873
Richardson, TX 75083

Via **WA5MWD.**

SOUTHNET PACKETCONFERENCE

The **SOUTHNET** packet Conference, cosponsored by The Florida Amateur Digital Communications Association **(FADCA)** and the Gator Amateur Radio Club **(GARC),** will be held at the University of Florida, in Gainesville, Florida on June 15 and 16, 1985. The conference will address a wide range of topics in

Gateway

Vol. 1, No. 18
Apr. 23, 1985

The ARRL Packet-Radio Newsletter

AUTOMATIC CONTROL OF DIGITAL COMMUNICATIONS

The FCC has responded to the ARRL petition for rule making that proposed allowing automatic control of digital communications on frequencies above 50 MHz. The response is PR Docket No. 85-105, Notice of Proposed Rule Making 4879, Parts of the NPRM are reproduced below.

"The Commission has received a petition from the ARRL, seeking to amend the Amateur Radio Service Rules to permit automatic control of digital communications on all amateur frequencies above 30 MHz. The ARRL notes that Part 97 currently contains provisions for automatic control of stations in repeater, auxiliary and beacon operation but makes no provision for automatic control of routine digital communications. In support of its petition, the ARRL states that a variety of digital [modes] such as radioteleprinter, transfer of computer programs, direct computer-to-computer communications and packet switching' systems lend themselves to a mode of amateur-radio transmission where a control operator need not be present. According to the ARRL, present microprocessor and computer technology now routinely present at amateur stations can automatically transmit and receive digital communications, verify receipt of messages and respond to inquiries. The ARRL notes that the use of Computer Based Message Systems (CBMS) are something new in amateur communications and should be encouraged by more experimentat ion, including automatic control which is both feasible and necessary to facilitate further development in the art of amateur radio. Two timely comments were filed. Both supported the petition for rule making.

"Automatic control in the Amateur Radio Service has previously been approved for repeater, auxiliary links and beacon operations. With an ever-growing list of amateur operations where automatic control is permitted, we believe that now may be the appropriate time to expand automatic control to all amateur operations, prohibiting its use only in those situations where there is a justifiable reason why automatic control should not be allowed. Therefore, we invite amateur radio operators in general, and amateurs experienced in automatic control in particular, to submit comments calling to our attention any problems that may arise by expanding automatic control to encompass all amateur radio operations. Our goal is to keep the amateur service abreast of technological developments and to utilize new technology, such as CBMS, where appropriate. On the other hand, we do not want to introduce any innovations into the service which would be disruptive of amateur communications or which would essentially change the character of the serv ice.

"We propose that any amateur radio station may be under automatic control, except when transmitting on frequencies below 29.5 MHz. As noted earlier, the petitioner did not request automatic control below 30 MHz. However, since automatic control is already permitted for repeater operation between 29.5 and 29.7 MHz, it is reasonable to make the lower limit for automatic control 29.5 MHz, rather than 30 MHz.

"These proposed rule amendments would still prohibit automatic control operation in any instance where the station is transmitting third-party traffic. This in in accord with Section 97.79 (d) of the amateur rules which specifies that a control operator must always be present when a third party is participating in amateur radio communications."

The complete text of the NPRM is available from the ARRL Information Services Department. Send an s.a.s.e. with 39 cents postage, and ask for RM-4879.

It is important that interested packet-radio operators, especially the operators of bulletin boards, file comments on both the intent and the implementation of this NPRM. The deadline for comments is June 25. A formal filing requires that you file an original and five copies of your comments and other materials. If you want each Commissioner to have a personal copy of your comments, send 11 copies. Please also send any formal or informal comments on the NPRM to the ARRL.

Ed.

SIMULATED EMERGENCY TEST

Steve Goode, K9NG, files the following report:

"On April 20 the Chicago Area Packet Radio Association (CAPRA) participated in a simulated emergency test (SET) in conjunction with the National Weather Service, the Northwest Illinois ARES and the Village of Hanover Park, Ill. The simulation stated that at 9:35 A.M. a tornado touched down in Hanover Park creating a 100-yard path of destruction running d iagona 1 ly through the village. An Emergency Command center was set up at the Hanover Park po 1 ice station under the command of the Village Chief of Police. Liaison officers 'to the police and fire departments and to the amateur radio net were under the chief's command. Initial weather

messages from one network to another, because there is no standard format for presenting subject, destination, source and distribution information in a message header. The committee agreed to study the X.400 series of message protocols and attempt to develop a subset of these standards that would meet the needs of amateur packet red io.

Prom **W4RI.**

PACKET FREQUENCIES

The CVRA-Southeastern Repeater Association, Inc. coordinates repeaters in North Carolina, South Carolina, Tenessee, Virginia and West Virginia. CVRA has established 145.01, 145.03, **145.05,** 145.07 and 145.09 as packet-radio frequencies.

From K4ARO.

REPRODUCTION OF GATEWAY MATERIAL

Material may be exerpted from **Gateway** without prior **permiss** ion, provided that the original contributor is credited and Gateway is identified as the source.

Gateway: The ARRL Packet Radio **Newsletter** is published **by the**

American Radio Relay League
225 Main Street
Newington, CT 06111

Jeffrey W. Ward, **K8KA**
Editor

Larry **E.** Price, **W4RA**
President

David Sumner, **K1ZZ**
Executive Vice President

Subscriptions are available to ARRL members and nonmembers, at the following rates:

For 25 issues of Gateway:
In the U.S.
ARRL Member $6.00
Nonmember $9.00
In **U.S.,** Canada, and Mexico by First Class Mail
ARRL Member $11.00
Nonmember $14.00
Elsewhere by Airmail
ARRL Member **$14.00**
Nonmember $17.00.

Prices are subject to change without notice. Remittance must be in U.S. funds, and checks must be drawn on a bank in the U.S.

Gateway: The ARRL
Packet Rad io News letter
American Radio Relay League
225 Main **St.**
Newington, CT 06111

Nonprofit Organization
U.S. Postage
PAID
Hartford, **Conn.**
Permit No. **2929**

"Phase I of WESTNET's 1985 plans is to install 145.01 MHz repeaters in southern California, This will remove the need for frequency translation, reducing the maximum number of digipeater hops to five, with only three hops between L.A. and San Francisco. This will result in an immediate improvement in network service.

"Phase If is to install multi-port digipeaters, 220-MHz radios and 9600-bit/s modems at each of the digipeater sites. Each digipeater will be linked to its neighbors at the higher speed and frequency, providing user access to that link via 145.01 MHz at 1200 bauds. This will avoid collisions between local traffic around each digipeater with traffic on its way to a more distant point. While not as good as a true network protocol, this will be much better than current, single-frequency linking.

"Since the multi-port digipeater is implemented on surplus Xerox 820 boards, WESTNET can switch to a true layer-three network when one becomes available. The Xerox 820 was designated as the target hardware for the first prototype layer-three network software.

"Once in place, WESTNET will allow a user to use the network as if it were a simple chain of digipeaters. The user will transmit and receive on 145.01 MHz. If he specifys more than one digipeater, his packet will be repeater between digipeaters at 9600 bauds on 220 MHz. The packet will be transmitted on 145.01 MHz by the last digipeater in the list, Standard digipeaters can be used on 145.01 MHz as far as necessary for a user to access a multiport digipeater site.

"It is expected that by 1986 and certainly by 1987, there will be a true network-layer protocol in place, and that the backbone network will move toward 1.2 GHz at 56 kbits/s. We also suspect that 145.01 network-access and digipeater capability will always be needed to handle entry-level users, as a redundant path for network resilience in case of emergency and to handle users in extreme outlying areas.

"It was agreed that the 220-MHz link frequency is 220.950 MHz. The modem standard is the K9NG 9600-bit/s modem. The controller is the Xerox 820 running the KE3Z multiport digipeater software, WESTNET technical coordination will take place on the W6IXU mailbox."

If you are interested in schematics of the K9NG modems, they are published in the proceedings of the Fourth ARRL Computer Networking Conference, The mulitport digipeater software is available for an 8" disk to Gateway at the ARRL. I hope that Harold's report provides some insight into how packet radio is changing in areas where the activity is overflowing a single 1200-bit/s channel.

Via DR NET.

EASTNET GROWING

The addition of several new stations to EASTNET has connected the network from Ottowa, Canada, to Washington, D.C. The "missing link" in northern New Jersey has been filled by WA2SNA-2, installed by the Ramapo Mt. Amateur Radio Club. Now, mail can be forwarded from the W3IWI MailBox in Washington, through WB2MNF in southern New Jersey, and on up the network to WA2RKN in Hyde Park, NY, W1AW in Newington, CT, or any of the WØRLI MailBoxes in the Boston, MA area. Because of high local activity at many places in the network, real-time, long-distance connections often are impossible or slow. WØRLI-type store-and-forward messages traverse the network relatively well, though .

The WØRLI MailBox and Gateway software for the Xerox 820 has is providing much of EASTNET (and beyond) with packet-radio message service. All of the following stations are linked in this store-and-forward network:

WØRLI - Westford, MA
K1BC - Lexington, MA
KE1G-1 - Goffstown, NH
N1DKF - Cranston, RI
KA1T - Harvard, MA
W1AW-4 - Newington, CT
WB1DSW - Manchester NH
WA2RKN-2 - Hyde Park, NY
WB2MNF - Medford, NJ
W3IWI - Clarksville, MD
KS3Q - Baltimore, MD
AK3P - Hummelstown, PA
K7PYK - Scottsdale, AZ
WA4SZK - Florence, SC

A message placed on any of these PBBSs for a user of any of the others will eventually be forwarded from node to node to the proper MailBox. Routing tables are maintained (by hand) at each node.

To the north, the the Plattsburgh Amateur Packet Radio Association, a division of the Champlain Valley Amateur Radio Club, has installed some stations that link the Mt. Ascutney, Vermont, digipeater to Ottowa, Ontario. This path allows EASTNET stations to access a BBS run by Wayne Bruce, VE3FXI. For those of you on EASTNET who want to try out the path to Ottowa, use WA1TLN-1, KD2AJ, W2UXC-1 and VE3PAK. All of these digipeaters are on 145.01 MHz.

via WØRLI, Bd.

DIGITAL COMMITTEE MEETING

The ARRL Ad Hoc Committee on Amateur Radio Digital Communication met at the Networking Conference in San Francisco, The most important issues discussed by the committee were protocol standards for TNC control possible standards for message format.

Doug Lockhart, VE7APU, is investigating the use of CCITT X.3, X.28 and X.29 protocols in amateur TNCse Doug's paper in the conference proceedings outlines the need for a standard set of commands and messages on amateur radio TNCse If TNC programmers used standard commands and messages, a single article could describe how to operate all amateur-radio TNCse If standards were adopted and adhered to, applications programs like the WØRLI MailBox would could use any TNC. The Committee expressed its appreciation for the research that Doug had done and requested that he deliver a draft of a proposed standard as soon as possible.

Hank Magnuski, KA6M, and several other PBBS operators discussed how difficult it is to move

We welcome the Packet Communicator to the growing list of commercially-available TNCs. The Kantronics packet-radio motto? "Packet Made Easy!"

Via Computers and Amateur Radio.

PACKET RADIO AND FIELD DAY

Due to a letter to Gateway from Thomas Clements III, W1ICH, the ARRL Contest Advisory Committee (CAC) has approved a Field Day packet-radio bonus. Field Day rules will now read: "An additional 100 points can be earned by completing at least one QSO on packet radio during the Field Day period. The repeater provision is waived for packet-radio QSOs. A packet station does not count as an additional transmitter. On the summary sheet, show packet radio as a separate 'band.'" This new bonus is in effect this year, thanks to quick action by the CAC.

Field Day, always a day of high visibility for Amateur Radio, is now a good time to show the public and the members of your Field-Day group the advantages of packet radio, Amateur Radio's newest mode. Show them that packet radio is no longer an experimental mode, but a valuable emergency communications tool. This Field Day bonus might also encourage some investigation of low-power portable packet stations. TNC, two-meter rig and portable computer might all be run from solar, wind or human power.

Mr. Clements suggests that "the annual Field Day message be sent to the section manager by packet radio if possible. Perhaps some traffic handling nets could arrange to pick up messages from packet-radio bulletin boards."

If you have any interesting ideas for using packet radio during Field Day, send them along to Gateway. Also, be sure to take some pictures of your portable packet station for the QST Field Dgy summary.

Ed.

TRANSCONTINENTAL PACKET

At the meeting of the Board of Directors of the Pacific Packet Radio Society (PPRS) on March 21st, 1985, the following resolution was passed:

Whereas the PPRS was one of the first societies formed specifically to encourage the growth of computer networking via radio using all digital concepts and techniques, and whereas the San Francisco area was the site of the nation's first amateur digipeater, and whereas an even greater challenge faces the amateur radio community to establish a transcontinental link, the Pacific Packet Radio Society has decided to establish a unique award to encourage the completion of the first terrestrial transcontinental network link. This one-time award shall be known as the "Golden Packet" [Like the golden spike that completed the transcontinental railroad. - Ed.] and the regulations relating to it are listed below:

1. A transcontinental link must be established, with each terminus located within 100 kilometers of either the Atlantic or Pacific Ocean.

2. The system must consist of fixed terrestrial digital store-and-forward radio links using VHF (greater than 144.1 MHz), UHF or microwave frequencies. Use of HF, satellite, tropo, meteor-scatter or moonbounce channels is prohibited.

3. A valid two-way transmission and acknowledgment of previously unknown information (256 characters or more> must occur in less than ten minutes.

4. This competition is open to licensed North American amateurs, and no commercial links or services may be used in the path. Club stations are permitted.

5. Proof of the exchange must be submitted to the PPRS. Proof must include a list of the stations in the link, their locations, frequencies used and a copy of the text exchanged.

6. The reward shall consist of a suitably engraved plaque with the names of all participating stations listed which shall be presented to the ARRL. Each participating station shall receive either a plaque or a certificate.

7. Final decision on the award is subject to review and approval by the Board of Directors of the PPRS.

From **KA6M.**

WESTNET MEETING

The second "official" meeting of WESTNET was held in conjunction with the Fourth ARRL Computer Networking Conference last week. The networking plans set forth at this meeting illustrate how most amateur packet-radio networks will advance. Several groups within EASTNET are making plans similar to the WESTNET plan. The following notes on WESTNET plans and the reasoning behind them come from Harold Price, NK6K.

"The primary purpose of the first WESTNET meeting was to put in place the best network possible with the technology then available. It was hoped that linking San Diego through Los Angeles to San Francisco would help generate interest at the endpoints and in the less populous areas between, and that the increased number of users would supply the resources to build a more sophisticated network. We are now ready to proceed with the first in a series of improvements: a 9600-bit/s network backbone.

"WESTNET decisions have been based on two concepts. First, use the best technology that is available, but don't wait for the next innovation that is coming 'real soon now.' Second, don't build anything that will make transfer to the next level of technology difficult.

"What technology is there to use now? There is still no agreement on network-layer protocols. There aren't even any working prototypes of network software. There is, however, a 9600-bit/s modem design, the ARRL lab's multi-port digipeater software, hundreds of users running 1200 bauds on two meters and a chain of seven digipeaters between southern and northern California. Unfortunately, a frequency translation from 145.01 MHz to 145.35 MHz is now necessary between north and south.

Gateway

Vol. 1, No. 17
Apr. 9, 1985

The ARRL Packet-Radio Newsletter

FOURTH ANNUAL NETWORKING CONFERENCE

The Fourth ARRL Amateur Radio Computer Networking Conference, cosponsored by the ARRL and the Pacific Packet Radio Society (PPRS), was a complete success. More than half of the twenty four papers published in the proceedings were presented at the conference, with attendance around 100 throughout the six hour meeting. The PPRS/ARRL booth at the West Coast Computer Faire, inc lud ing on-the-a ir demons trat ions of packet radio at 1200 and 9600 bauds, generated a lot of interest. Several hundred handouts were given to interested Faire-goers, and a lot of radio amateurs who didn't know about packet radio stopped by the booth to see the TNCs in action. Congratulations to Hank Magnuski, KA6M and the PPRS for coordinating a successful conference.

The Proceedings of the Fourth ARRL Amateur Radio Computer Networking Conference are available from the ARRL for $10. See Gateway issues 14 and 15 for abstracts of the papers of the papers is this 109-page volume of proceedings. Proceedings for the first, second and third conferences are also available for $8, $9 and $10, respectively. These publications provide a valuable record of the history of packet radio, what packet-radio experimenters expect from the future and diverse points of view on many technical issues.

TRENTON COMPUTERFEST

For those on the East Coast, there will be a large packet-radio meeting at the Trenton Computerfest, on April 20th.

1000 Opening remarks - Harold Winard, KB2M.
1010 ARRL remarks - Jeff Ward, K8KA.
1020 Introduction to packet - Jon Pearce, WB2MNF.
1115 Regional summaries of packet activity.
1245 Comparisons of packet-rad io TNCs.
1345 Introduction to the WØRLI MailBox - Dick Kutz, KS3Q.
1430 Packet-radio expert-panel discussion with Tom Clark, W3IWI, Phil Karn, KA9Q and Mike Bruski, AJ9X.
1600 End of forum.

The room-number for the forum is not available at this time; just ask at the Computerfest administration booth or look for posters. If you would like to make a presentation during the "regional summaries," contact Jon Pearce, WB2MNF, at 609-953-1566, or Harold Winard, KB2M, at 201-361-6478.

via WB2MNF.

DAYTON HAMVENTION

Packet-radio will be well represented at the Dayton Hamvention, April 26, 27 and 28. The Dayton packet forum will be held on Friday, April 26th from 2:30 P.M. until 5:30 P.M., in room 1. Six speakers will make presentations and answer questions from the audience.

1430 Introduction to the packet forum - Bob Neben, K9BL
1440 Telecommunications and the Amateur in the 21st Century - Rick Whiting, WOTN.
1500 Packet Primer, 1985 - Pete Eaton, WB9FLW.
1530 Report on the ARRL and packet-radio technology - Jeff Ward, K8KA.
1600 Update on TAPR - Lyle Jonhson, WA7GXD.
1630 PACSAT update - Harold Price, NK6K.
1700 Networking, bulletin-board systems, and the Xerox 820 computer - Terry Fox, WB4JFI.

There will also be display booths for each of the major packet radio TNC manufacturers, including TAPR, GLB, Heathkit, AEA and Kantronics.

Packeteers are encouraged to use 145.01 MHz (voice) as a coordination frequency.

Via K9BL.

KANTRONICS TNC

Kantronics, makers of several RTTY and AMTOR software/hardware packages, have announced the availability of a TNC, the Kantronics Packet Communicator. An early production model of the Packet Communicator was displayed at the recent Computer Networking Conference in San Franc isco. The TNC is small, housed in a 2 X 6 X 8-inch cabinet, much like the cabinet for the Kantronics UTU.

Although Kantronics "took some cues from the TAPR TNC," the Packet Communicator is a new hardware design. The internal modem uses switched-capacitor filters and can be switched by software to provide either Bell 103 or 202 tones, with or without receiver equalization. Also, the Packet Communicator can be used as a Bell 202 modem, bypassing all packet-radio functions. The TNC can send and receive packets at 300, 400, 600 and 1200 bit/s, half duplex.

The Packet Communicator serial port can provide RS-232C or TTL signals, at 300, 1200 or 9600 bauds. Special packet-radio terminal programs for many computers will be available soon from Kantronics.

TRENTON COMPUTER FEST

There will be a packet radio meeting at the Trenton (New Jersey) Computer Fest on Saturday, April 20 (note the change -- some of the earlier publicity had indicated Sunday). The Mid-Atlantic Packet Radio Council (MAPRC) is sponsoring the packet sessions and the organizers are Harold Winard, KB2M and Jon Pierce, WB2MNF. There will be a panel of experts to answer questions about packet radio and presentat ions by representatives of various EASTNET LANs.

INTERNATIONAL PACKET ACTIVITY

Although this report and list were biased toward U.S. packet activity, there is considerable activity in Canada and overseas. Japan, the U.K., New Zealand, Australia, Germany and Sweden all have packet radio activity. Canadian activity-centers include Hamilton, ON; Vancouver, BC; Toronto, ON; and Ottowa, ON. Back issues of Gateway provide more informat ion on international activity.

REPRODUCTION OF GATEWAY MATERIAL

Material may be exerpted from Gateway without prior permission, provided that the original contributor is credited and Gateway is identified as the source.

Gate-way: The ARRL Packet Radio Newsletter is published by the

American Radio Relay League
225 Main Street
Newington, CT 06111

Jeffrey W. Ward, K8KA
Editor

Larry E. Price, W4RA
President

David Sumner, K1ZZ
Execut ive Vice President

Subscriptions are available to ARRL members and nonmembers, at the following rates:

For 25 issues of Gateway:
- In the U.S.
  - ARRL Member $6.00
  - Nonmember $9 .OO
- In U.S., Canada, and Mexico by First Class Mail
  - ARRL Member $11 .OO
  - Nonmember $14.00
- Elsewhere by Airmail
  - ARRL Member $14.00
  - Nonmember $17.00.

Prices are subject to change without notice. Remittance must be in U.S. funds, and checks must be drawn on a bank in the U.S.

Gateway: The ARRL
Packet Radio Newsletter
American Radio Relay League
225 Main St.
Newington, CT 06111

Nonprofit Organization
U.S. Postage
PAID
Hartford, Conn.
Permit No. 2929

MASSACHUSETTS

New England Packet Radio Assn.(NEPRA)
P.O. Box 15
Bedford, MA 01730

MICHIGAN

Eastern Packet Radio Of Michigan (EPROM)
c/o J. Nugent, WB8TKL
307 Ross Dr.
Monroe, MI 48161

MINNESOTA

MAPR hosts a Tuesday-evening voice net on 146.04/64 MHz at 7:45 PM. Packet operation is on 145.010 MHz with additional channels on 145.3, 145.5, 145.7, and 145.9 MHz.

Minnesota Amateur Packet Radio (MAPR)
C/O Pat Snyder, WAØTTW
565 Redwood Lane
New Brighton, MN 55112

NEW HAMPSHIRE AND **VERMONT**

Mt. Ascutney Amateur Packet Radio Association
c/o Carl Breuning, N1CB
54 Myrtle St.
Newport, NH 03773

NEW JERSEY

0 The Radio Amateur Telecommunications Society provides support to amateurs engaging in packet activities. The group's activities include a packet software library and development of a cross-state trunking system to be incorporated into EASTNET. A directory of active amateur packet stations and facilities has been compiled and is available. Standards documents are kept in club files and may be distributed upon request. RATS is developing an interface between the private teleconference DR NET and the EASTNET packet-radio network. This interface should be operat ing soon.

In Northern New Jersey/New York City:
The Radio Amateur Telecommunications Society (RATS -NORTH)

c/o J. Cordon Beattie, Jr. N2DSY
206 North Vivyen St.
Bergenf ie ld, NJ 07621
(201) 387-8896

In South Jersey:
The Radio Amateur Telecommunications Society (RATS-SOUTH)
c/o Brian B. Riley, KA2BQE
RD 2 Burnt House Rd.
Indian Mills, NJ 08088

RATS BBS Tel: 609-268-9597 (300 baud)

0 The Cherryville Repeater Association is also active in packet radio in New Jersey. Their main concerns are public service and 220-MHz linking. They maintain a series of linked 220-MHz duplex repeaters that are used for voice and packet.

Cherryville Repeater Association
Box 308
Quakertown, NJ 08868

NEW YORK

Rochester Packet Group
c/o Fred Cupp, W2DUC
27 Crescent Rd.
Fairport, NY 14450

0 New York City:

Packet Of New York (PONY)
c/o Bill Schimoler
42-15 172 St
Flushing, NY 11358

0 Hudson Valley:

The Mount Beacon Amateur Radio Club supports an active packet-radio group. They are now turning their attention and resources to 220-MHz network links.

Mt. Beacon Amateur Radio Club
P.O. Box 841
Wappingers Falls, NY 12590

OHIO

Cincinatti Amateur Packet Radio Experimenters Society (CAPRES)
c/o John Schroer-IV, KA8GRH
984 Halesworth Dr.
Forest Park, OH 45240

Cleveland Area:
Maynard Weston, W8MW
4564 Park Edge Dr.
Fairview Park, OH 44126

TENNESEE

Tennesee packet activity is centered in Memphis, on 145.01 MHz. Contact:

John Burn ingham, WB8PUF
Memphis State University
Dept. of Engineering , Technology
Memphis, TN 38152

TEXAS

Dave Cheek, WA5MWD
1510 Treavis St.
Garland, TX 75042

UTAH

UPRA has a voice net on Tuesday evenings, at 8:45 PM, on 146.02/62 MHz.

Utah Packet Radio Association (UPRA)
4382 Cherry-view Drive
West Valley City, Utah 84120

WASHINGTON

Northwest Amateur Packet Radio Association (NAPRA)
c/o John Gates, N7BTI
750 Northstream Ln.
Edmonds, WA 98020

While Sacramento packet stations are able to get into the San Francisco Bay Area with moderate success, plans are to install a high-level digipeater in the Sierra Nevada this year which will allow for more reliable links into the Bay area and down the Coast Range. Even better, the digipeater is planned to be a major node in the proposed Central Valley chain which will provide North-South path through the center of the state. SACPAC is also working on a WØRLI MailBox in Sacramento which should be functioning on 145.09 MHz by early summer. SACPAC is coordinating with PPRS to ensure that operating parameters remain standardized not only between these groups, but also within the rapidly expanding group of newcomers to packet radio in this area. All local operations are on 145.01 and 145.03 MHz.

The Sacramento County Office of Emergency Operations plans to have a TNC installed by the end of this summer and is coordinating with the State Office of Emergency Services to establish links and work out details for use of packet systems in emergency and disaster situations. The plan is to have packet "teams" within the County's newly-revitalized RACES program and ARES. The teams would set up communications at appropriate sites during drills and actual emergenc ies. A good framework should be developed over the next year to accomodate the projected growth in this emergency network.

0 LAPG is the local Los Angeles packet-radio club. LAPG
8:00 PM local time on 145.36 MHz simplex.
of every month

Los Angeles Area Packet Group (LAPG)
P.O. Box 6026
Mission Hills, CA 91345

0 The Southern California Digital Coordination Council (SCDCC) is an organization formed by the packet radio operators in Southern California to serve as a central point for communications between packet users and other amateur groups. It also serves as a clearing house for information on local packet activity. Membership is open to anyone; packet radio users are encouraged to join. SCDCC has a technical committee to help track and promote the rational growth of packet radio equipment, repeaters, gateways and protocols on frequencies assigned to packet radio.

SCDCC serves all of Southern California and has members in San Diego, Los Angeles, Ventura, Santa Barbara, Lancaster, Lompoc, and other areas.

SCDCC
P.O. Box 6026
Mission Hills, CA 91345

0 The Pacific Packet Radio Society was one of the first North American packet-radio clubs. PPRS serves the San Francisco Bay area, and is the co-sponsor of the Fourth Amateur Radio Computer Networking Conference.

Pacific Packet Radio Society
P. 0. Box 51562
Palo Alto, CA 94303

0 There is also a packet-radio club in San Diego.

San Diego Packet Group (SDPG)
c/o Mike Brock, WB6HHV
10230 Mayor Circle
San Diego, CA 92126

COLORADO

Rocky Mountain Packet Radio Association
% Andy Freeborn, NØCCZ, Secretary
5222 Borrego Drive
Colorado Springs, CO
(303) 598-8373

FLORIDA

The Florida Amateur Digital Communications Association is an organization interested in digital communication techniques such as packet radio. Since its founding in 1983, FADCA has grown to be one of the major regional packet organizations in the nation. FADCA members are from Florida, Geoıgia, and many other states. FADCA provides a structure for planning orderly growth of packet networking, and has been designated as the agent of the Florida Repeater Council for administering packet frequencies and repeater sites. FADCA's newsletter, the BEACON, provides technical information and operating news to over 250 persons each month. FADCA is the coordinator of the SOUTHNET packet network which is rapidly expanding from Florida through the entire Southeast.

Florida Amateur Digital Communications Association (FADCA)
812 Childers Loop, Brandon, FL 33511
(813) 689-3355

GEORGIA

Georgia Radio Amateur Packet Enthusiast Society
GRAPES
P.O. Box 1354
Conyers, GA 30207

Southern Amateur Packet Society (SAPS)
c/o Wayne Harrell, WD4LYV
RT 1 Box 185
Sycamore, GA 31790

ILLINOIS

Chicago Amateur Packet Radio Assn. (CAPRA)
P.O. Box 8251
Rolling Meadows, IL 60008

St. Louis Area Packet Radio
9926 Lewis & Clark
St. Louis, MO 63136

IOWA

The Central Iowa Technical Society has helped build a widespread packet network throughout Iowa.

Central Iowa Technical Society
c/o Ralph Wallio, WØRPK
RR 4
Ind iano la, IA 50125

KANSAS

John Anderson III, WBØSKL
305 Brittany
Olathe, KS 66061

Gateway

Vol. 1, No. 16
Mar. 26, 1985

The ARRL Packet-Radio Newsletter

SPECIAL ISSUE

This issue of Gateway is tailored to provide information to those attending the West Coast Computer Faire and the Fourth Amateur Radio Computer Networking Conference. We have attempted to provide an overview of packet-radio activity in the United States and a fairly complete list of U.S. packet-radio clubs.

PACKET RADIO IN 1985

The past year has seen packet radio grow from a few hundred isolated stations into a few dozen tentatively-connected local-area networks (LANs). There are now between 2,000 and 3,000 packet-radio terminal node controllers (TNCs) operating 'on the amateur bands. The following list of clubs gives a fair idea of where the activity is concentrated: The San Francisco Bay area and the Los Angeles basin are the hotspots of "WESTNET" activity. These two centers are linked by a string of mountaintop repeaters. In the Northwest, there are organized groups pushing packet radio from Vancouver, BC toward Northern California. There are two paths taking shape from West to East: Arizona - New Mexico - Texas - Arkansas, and Utah - Colorado - Nebraska - Iowa - Illinois. The Chicago-area has recently produced a 9600-bit/s modem (designed by Steve Good, K9NG) and promises to be a center of network development. The east coast from Washington, D.C. to Boston is tenouosly linked, and there are many areas of high activity along this "EASTNET." In the south, Florida has the largest packet-radio population, and Alabama and Georgia are beginning to come on-the-air. Terminal node controllers have been sold in every state in the U.S. and in several overseas countries.

The U.S. is being covered by LANs, and the next leap for amateur packet radio will be to link these LANs using a "network-layer protocol." Many LANs are already linked in a store-and-forward mail network designed by Hank Oredson, WØRLI. Hank's MailBox software, running at dozens of sites throughout the world, forwards a message from one node to another, until the message reaches its destination. This taste of networking is driving several software and hardware teams in the development of real-time networking equipment.

If you are excited by the idea of an amateur data network, or if you have some use for such a network (emergency traffic, weather networking, message handling, etc.) contact a packet-radio group and join the fun.

Jeff Ward, K8KA, Editor.

NATIONAL ORGANIZATIONS

- The Amateur Radio Research and Development Corp. is a group interested in advanced amateur-radio techniques. Members of AMRAD have been responsible for many advances in packet radio.

AMRAD
P.O. Drawer 6148
McLean, VA 22106-6148

- The Amateur Radio Satellite Corp. has ongoing interest in the development of amateur packet-radio standards, and they are currently involved in the construction of PACSAT, a packet-radio satellite.

AMSAT
850 Sligo #601
Silver Spring MD 20910
301-589-6062

- The Tucson Amateur Packet Radio Corp. (TAPR) is best known for the TNC kit that it designed, tested and markets. With the TNC now being produced by two commercial companies, TAPR continues to be instrumental in packet-radio development .

Tucson Amateur Packet Radio (TAPR)
P.O. Box 22888
Tucson, AZ 8573402888

ALABAMA

In the Huntsville area, contact:

Frank Emens, W4HFU
3714 Lakewood Circle
Huntsville, AL 35811.

Near Birmingham:

Henry Wingate, K4HAL
104 Von Dale Drive
Birmingham, AL 35215

ARKANSAS

Elmer Wingfield, W5FD
26 Belmont Drive
Little Rock, AR 72204

CALIFORNIA

- Recently, the Sacramento Packet Group (SACPAC) became a special interest group under the Sacramento Amateur Radio Club, Incorporated. This club, W6AK, is the area's oldest ham club.

If you live in the west, consider attending the Rocky Mountain Division Convention in Jackson Hole, Wyoming, August 2 through **4.** There is a lot of packet activity planned for the Convention, including a presentation by Pete Eaton, **WB9FLW.**

Prom **N7BHC.**

MORE READING

"Designing and Implementing Local Area Networks," by Dimitris N. Chorafas, McGraw Hill, 1984. This book provides a good overview of **LANs.** Not all of the discussion involves packet radio, but many interesting topics are addressed.

FROM K1HOP.

REPRODUCTION OF GATEWAY MATERIAL

Material may be exerpted from **Gateway** without prior permission, provided that the original contributor is credited and Gatewav is identified as the source.

Gateway:. The ARRL Packet Radio Newsletter is published **by the**

American Radio Relay League
225 Main Street
Newington, CT 06111

Jeffrey W. Ward, **K8KA**
Editor

Larry E. Price, **W4RA**
President

David Sumner, **K1ZZ**
Execut ive Vice President

Subscriptions are available to ARRL members and nonmembers, at the following rates:

For 25 issues of Gateway:
- **In** the U.S.
  - ARRL Member $6.00
  - Nonmember $9.00
- In U.S., Canada, and Mexico by First Class Mail
  - **ARRL** Member $11.00
  - Nonmember $14.00
- Elsewhere by Airmail
  - ARRL Member **$14.00**
  - Nonmember $17.00.

Prices are subject to change without notice. Remittance must be in U.S. funds, and checks must be drawn on a bank in the U.S.

Gateway: The ARRL
Packet Radio Newsletter
American Radio Relay League
225 Main St.
Newington, CT 06111

Nonprofit Organization
U.S. Postage
PAID
Hartford, **Conn.**
Permit No. 2929

"Another Application Note Describing a Low-Power RS232-Like Interface," by Paul Newland, AD71 -- A new and improved low power RS232-like interface is discussed. It features half the power consumption of the interface presented in a previous paper, and it uses fewer parts.

"A Few Thoughts on User Verification Within a Party-Line Network," by Paul Newland, AD71 -- this paper presents an idea for verifying&that a user within a party line network is who he or she claims to be. The idea assumes that the channel is a party line and the potential intruders will monitor authorized communications and may attempt to masquerade as authorized users. No attempt is made to encrypt the authorized user's data for transmission over the party line.

"A More Watchful Watchdog for Microcomputers," by Paul Newland, AD71 -- Many hardware/software watchdog timers consist of a software routine that repetitively triggers a hardware retriggerable monostable. If the monostable ever times out, the computer is reset. This technique, although useful, is not extremely reliable under most software/hardware insane conditions. This paper discusses an alternative approach that may prove to be more reliable under various fault conditions.

You can see from these abstracts that many interesting papers will be presented at the conference (and published in the proceedings). Whether you are interested in learning more about packet-radio protocols, applications or equipment, you will be excited by the conference. If you think that you will be left behind by the technical papers, be sure to attend "Pete's Packet Primer," which will begin at 10:45 AM on March 30th in room 232 (the same room that will house the rest of the conference).

Details of the conference remain as stated in Gateway issue 14, with the following additions: On March 29th, beginning at 8:00 P.M., there will be a Network Management meeting at the San Francisco Pizzeria, 418 Beach Street. On March 30th (the day of the conference) there will be an 8:30 breakfast at Tad's restaurant, 12 Powell St., and an 8:00 P.M. dinner at New Joe's Restaurant. Dinner reservations will be taken during the conference. If you have any questions about the Amateur Radio Computer Networking Conference, contact Hank Magnuski at (415)-854-1927. If you have any questions about the Computer Faire itself, call (415)-364-4294.

Via **KA6M**, Ed.

SPECIAL CONFERENCE ISSUE OF GATEWAY

Issue number 16 of Gateway will be distributed at the West Coast Computer Faire and the Fourth ARRL Amateur Radio Computer Networking Conference. This will be a special issue, providing the casual reader with an overview of packet radio and current packet activity and a list of packet-radio clubs. Even if you think that we already have information on your club, contact Gateway to be sure that you are included in this special issue. Include a description of any projects that your club is involved in. Remember, several hundred copies of Gateway number 16 will be handed out at the Faire, this is your chance to get some for your club. Leave the call or write ARRL Headquarters.

Ed.

EMERGENCY COMMUNICATIONS

Pattie Winter, N6BIS, will be writing a major article on the use of packet radio in emergency communications, and she is seeking information from anyone who has experience in this area. If you have written an article on this subject or done emergency planning which incorporates the use of packet-radio technology, please send a copy of the article to Patti at

Patti Winter, N6BIS
P.O. Box 537
Menlo Park, CA 94026.

Patti would also like to meet and interview anyone who has used packet radio for emergency communications. She will be holding interviews at the Fourth Amateur Radio Computer Networking Conference.

From KA6M

GEORGIA LIAISON

The Georgia Radio Amateur Packet Enthusiast Society (GRAPES) now has a "packet-radio liaison," responsible for sending out information packages about packet radio and keeping other organizations up to date on packet radio in Georgia. People interested in the information package should send $1.00 and an s.a.s.e. to

Bill Crews, WB2CPV
1421 Hampton Ridge Road
Norcross, GA 30093.

From **WB2CPV.**

HF ACTIVITY

While there is considerable packet activity on the HF bands, it's hard to say when or on what frequencies. To help ease this confusion, paving the way for more HF packet activity, Gateway, with the help of Pete Eaton, WB9FLW, is compiling a list of active HF packet stations, the frequencies that they use, and the times that they are most active. If you are on HF packet radio, send information about your activity to Gateway, at ARRL Headquarters.

WB9FLW, Ed.

INTERMOUNTAIN NETWORK

Dave Pedersen, N7BHC, from Salt Lake City, Utah, will be travelling to Boise, Idaho, hoping to spread packet radio in the west. Dave thinks that the 30 or 40 interested hams in Boise could be linked to the growing network in Salt Lake City by midsummer. Rod Greeene, W7ZRC, in Boise feels that extending the link from Boise to Pullman, Washington and then to Seattle would be fairly easy.

transmit timely forecast information from the Hurricane Center to effected areas.

"Packet Radio for Distance Teaching in the Third World," by Phil Gray, KA7TWQ -- This paper discusses how packet radio could improve the interactive capabilities of "Distance Teaching" (delivery of education to remote or sparsely-populated areas>. It also presents views on how packet radio could improve the delivery and efficiency of Distance Teaching in Less-Developed Countries.

"EASTNET: A Year Later," by Bob Bruninga, WB4APR -- The original goal of EASTNET to link the eastern seaboard from Washington, DC to Boston was met, more or less, on August 6, 1984, when packets were exchanged between Elk Neck, Maryland, and Lowell, Massachusetts. Since that time, numerous alternate paths have been exercised. The saturation of the primary link frequency of 145.01 MHz during prime evening hours, however, has prevented routine end-to-end, multihop paths. This paper presents some methods of relieving this congestion.

"Activity Report of [Japan's] PARNET," Yamazaki, et al -- We have been studying and working on digital communications in ham radio for the past year. The group's activities are described in this paper. In section one, there are members' profiles, the group's objectives and the results of our investigations. In section two, the hardware of the prototype PARNET TNC is described. In section three, the TNC software is described, and prospects for the future are presented.

"Formal Definition Meeting for the Packet Radio Experiment (RUDAK) to be Included in AMSAT P3-C," by Karl Meinzer, DJ4ZC and Hans Peter Kuhlen, DK1YQ -- During the weekend February 15 through 17, AMSAT-DL hosted a formal meeting to define the packet payload in P3-C. The experiment has been named "RUDAK" for "Regenerativer Umsetzer fur Digitale Amateur-Kommunikation" [Loosely: "regenerative transponder for amateur digital communication." -- Ed.]. This paper presents the resulting design, including link budgets, modulation schemes, and bit rates.

"Of Virtual Circuits, Datagrams, and the Circular File," by Terry Fox, WB4JFI -- Even as work was being completed on the link layer, amateurs were beginning to take on the challenge of designing a true amateur packet network. Two "camps" have taken shape in this stage of development work: the "virtual circuit" camp and the "datagram" camp. This paper presents a slightly biased view of these two camps and their protocols.

"FADCA GATOR LINK 1 Packet Radio Linking Network," by Howard Goldstein, N2WX and Ted Huf, K4NTA -- The GATOR LINK 1 concept was devised in the summer of 1984 by a group of members of the Florida Amateur Digital Communications Association (FADCA) as a method of linking packet-radio digipeaters into a system that would provide wide-area communications without the problems involved in single-frequency digipeating. It was recognized that, while AX.25 Level 2 provided a means of linking digipeaters, as packet radio activity grew, it would become more difficult to use this feature because of collisions. The GATOR LINK 1, as described in this paper, provides a solution to this problem.

"Communications Protocols for the Network and Transport Layers of the Amateur Packet Network," by Gordon Beattie, Jr., N2DSY -- There has been much discussion among amateurs about internetworking with other areas of the country and globe. This has lead to introduction of terms into the vocabulary of many amateurs, many of whom are newly equipped with computers! In this paper, we will present the ISO/CCITT Open Systems concept and its impact on the protocols that we wish to use to provide reliable data transfer in the amateur network. In order to provide a basis for contrast, we will also introduce the U.S. Department of Defense model of communications systems.

"TCP/IP: A Proposal For Amateur Packet Radio Levels 3 and 4," by Phil Karn, KA9Q -- This paper presents a case for basing Level 3 (the network layer) of amateur packet radio on the "datagram" concept. It further proposes that the DARPA protocols IP (Internet Protocol) and TCP (Transmission Control Protocol) be adopted intact as the standard Level-3 (Network) and Level-4 (transport) protocols for Amateur Packet Radio. I will then provide an overview of TCP/IP, explain why it, as a datagram protocol, is more suitable for our needs than the virtual-circuit protocol CCITT X.75, and show how it would be used above the AX.25 Level-2 protocol already in use.

"Addressing and Routing Issues in Amateur Packet Radio," by Phil Karn, KA9Q -- As amateur packet radio evolves from scattered, ad-hoc collections of local digipeaters into a large, automatic and interconnected network, several issues related to naming, addressing and routing will have to be faced and overcome. Routing, in particular, has long been a fertile research area in computer networking. I make no claim to knowing the answers to many of these problems; however, I believe that they can at least be stated, and that certain decisions can be made early to ease experimentation with various solutions. In particular, the problem of address assignment is discussed with particular emphasis on making the routing problem easier.

"Proposal: Recommendation AX.121NA, a Numbering Plan for the Amateur Radio Network in North America," by Gordon Beattie, Jr., N2DSY -- The purpose of this numbering plan is to facilitate the introduction of amateur data networks and provide for internetworking in North America.

"CCITT X.244 Transport Layer Protocol Basic Description," by Terry Fox, WB4JFI -- In order to assure absolute data integrity through the amateur network, some form of transport-layer protocol should be employed between the entry and exit points of the network. This paper proposes the use of another CCITT X-series protocol to correct for potential deficiencies in AX.25 Level 3.

"X.3 and X.28 Protocols for Terminal Node Controllers," by Douglas Lockhart, VE7APU -- This paper proposes the adoption of an extended version of CCITT recommendations X.3 and X.28 for use in amateur-radio terminal node controllers (TNCs). The various X.3 parameters and X.28 commands and service signals (messages) are outlined and the extensions in place in the V-2 software implementation on the Vancouver Amateur Digital Communication Group (VADCG) TNC are discussed.

Gateway

Vol. 1, No. 15
Mar. 13, 1985

The ARRL Packet-Radio Newsletter

PACSAT MEETING

I had the good fortune to attend the design meeting for PACSAT held recently in Rosslyn, Virginia. At this meeting, representatives of AMSAT, the Volunteers In Technical Assistance (VITA) and NASA discussed the details of PACSAT funding, design and construction.

The most important issue discussed at the meeting was funding. Like all amateur satellites, PACSAT will be inexpensive, but sources of major funding will still have to be identified. PACSAT represents a unique opportunity for funding organizations to take part in a technological experiment that is also a social experiment; PACSAT, if successful, will prove that low-orbit data-communications satellites can be used to coordinate development and relief efforts in Third-world countries. Of course, PACSAT will also increase the utility of amateur packet radio by providing world-wide packet forwarding. If your group would like to contribute to PACSAT, or if you can help identify sources of major funding, contact

Dr. Gary Garriott
VITA PACSAT Project Officer
1815 North Lynn Street
Arlington, VA 22209.

After discussing funding, the group concentrated on technical considerations. Operators will see PACSAT as a packet bulletin board system (PBBS) in low-earth orbit. The PBBS will have somewhere between 2 and 4 megabytes of memory, and you will be able to access the satellite with an AX.25 TNC and some special radio/modems. This is all relatively straightforward, when compared to other aspects of PACSAT's design. AMSAT and VITA hope to have PACSAT launched as a Getaway Special (GAS) payload on the space shuttle. Such a launch places complex constraints on PACSAT's size, shape, propulsion system and construct ion. PACSAT must fit in a standard "GAS can" which is only 19 inches in diameter and 20 inches tall. Fitting several megabytes of memory, spacecraft controllers, transmitters, receivers, batteries, and a propulsion system into less than 5 cubic feet will be quite a job. To protect the shuttle and its crew, PACSAT will have to go through safety and construction tests more complex than AMSAT has ever confronted. The successful construct ion, launch and operation of PACSAT will depend on and demonstrate the best that Amateur Radio has to offer. That this is also a packet-radio project makes it of interest to all Gateway readers.

Ed.

PACKET PAPERS

Several more papers for the Fourth ARRL Amateur Radio Computer Networking Conference have arr ived. Here are the abstracts for those papers:

"The Frequency Agile Message System (FAMS)," by Dave Borden, K8MMO -- With the increasing traffic appearing on local-two meter packet radio channels, computer message systems appear to the casual conversationalist as channel hogs. The message systems interfere in two modes; First, a typer user connects with it and downloads large packets of data, help messages and directories. Second, a semi-automatic store-and-forward mode is invoked periodically to forward messages to other systems further up the network. The service that these message systems provide more than justifies their existence, so the answer is not to ban them. New operating practices might alleviate the interference and retain the useful features of the bulletin board systems. This method involves frequency agility -- the ability to change frequencies on command.

"Packet Radio Development - 1985," by Lyle Johnson, WA7GXD -- A review of packet growth since the Third ARRL Networking Conference is followed by a discussion of anticipated expansion of packet activity during the next year. A framework for orderly growth is presented, based on the above observations.

"The Realities of Packet Radio in the Amateur Radio Service, circa 1985, or How to Deal with a User Base," by Harold Price, NK6K -- The author postulates the existence of two major experimenter groups in amateur packet radio: those who experiment with data sent via packet radio and those who experiment with the way data is sent via packet radio. The problems of these groups in the face of 5000 or more packet users by the time of the 5th ARRL Computer Networking Conference are discussed.

"The Implications of Traditional Operating Practices for Amateur Packet Network Design," by Gwyn Reedy, W1BEL -- Amateur packet network design relies heavily upon equipment and procedures developed for commercial service. This paper urges network designers to examine the anticipated amateur uses of the network and modify commercial practice to accommodate the existing amateur population.

"Packet Radio and the National Hurricane Center," by Joel Kandel, KI4T -- Amateur Radio operators in South Florida are exploring the ability of packet radio to provide the National Hurricane Center with high quality weather observations and then to

HAMNET is a well-run discussion of all topics in Amateur radio. Section 9 in HAMNET is devoted exclusively to packet radio, RTTY, and AMTOR. Messages and news items for Gateway can be directed to CompuServe account #75105,737.

Ed.

RMPRA POWER SUPPLY DATA

Not all of the surplus power supplies that are "just like the Radio Shack" are suitable for conversion to power TAPR TNCs. To be safe, buy only the supplies specified in the Rocky Mountain Packet Radio Association (RMPRA) documentation. This documentation is available from RMPRA for an s.a.s.e. to:

RMPRA Power Supply Experiment
3775 East 115 th Ave.
Thornton, CO 80233.

Ed.

REPRODUCTION OF GATEWAY MATERIAL

Material may be exerpted from Gateway without prior permission, provided that the original contributor is credited and Gateway is identified as the source.

Gateway: The ARRL Packet Radio Newsletter is published by the

American Radio Relay League
225 Main Street
Newington, CT 06111

Jeffrey W. Ward, K8KA
Editor

Larry E. Price, W4RA
President

David Sumner, K1ZZ
Executive Vice President

Subscriptions are available to ARRL members and nonmembers, at the following rates:

For 25 issues of Gateway:
In the U.S.
ARRL Member $6.00
Nonmember $9.OO
In U.S., Canada, and Mexico by First Class Mail
ARRL Member $11.OO
Nonmember $14.00
Elsewhere by Airmail
ARRL Member $14.00
Nonmember $17.00.

Prices are subject to change without notice. Remittance must be in U.S. funds, and checks must be drawn on a bank in the U.S.

Gateway: The ARRL
Packet Radio Newsletter
American Radio Relay League
225 Main St.
Newington, CT 06111

Nonprofit Organization
U.S. Postage
PAID
Hartford, Conn.
Permit No. 2929

"Computer Networking in Japan...," by Robert Richardson, W4UCH -- The evolution, development, and implementation of the Microsoft MSX operating system in nearly 2 dozen models of microcomputers now being manufactured in the Far East and Pacific is discussed. The paper also addresses the impact of these MSX computers on packet radio on the amateur bands, primarily in Japan, for 1985 and onward.

"AX.25 Net Operation in the Connected Mode Using the Software Approach," by Robert Richardson, W4UCH -- This brief paper presents the means whereby an amateur radio net may be conducted using the AX.25 packet protocol with all stations in the connected mode. Use of a net control station connected simultaneously to all members of the net is described. The paper also discusses window overlays on video displays of all members of the net to display other net members' packet information fields.

If you have written a paper for the conference, the deadline has passed, rush your paper to Marian Anderson, WA1FSB, at ARRL Headquarters.

Via KA6M, Ed.

PUBLICITY IN JAPAN

The January 1985 issue of CQ HAM RADIO, a thick, glossy Japanese ham radio magazine, contains a 2-page spread covering the TAPR TNC. The article, written by JH3XCU, mentions TAPR, VADCG and AX.25. Considering the fact that this magazine has a circulation of 500,000, this article should stimulate significant interest in packet radio.

From **W3IWI.**

XEROX 820S

You may have noticed that a lot of packet-radio software has been developed for the Xerox 820 computer. These computers have become a favorite of the U.S. packet community. Now they are spreading throughout the world. Miki Nakayama, JR1SWB, coordinator of the packet portion of the JAS-1 satellite project, has a couple of Xerox boards, and he intends to run the WORLI MailBox on one of them. ZL1AOX, in Aukland, New Zealand is also using a Xerox and the WORLI software.

[The Xerox 820 is a Z-80 computer with 64 kbytes of RAM, on-board video, and serial and parallel I/O. Best of all, the boards are usually available from the Xerox Outlet for less than $100. Call the Xerox Outlet (persistently) at (214)-960-3367. -- Ed.]

From **W3IWI.**

SOFTWARE TNCS

People getting interested in packet radio often ask, "Why can't I program my computer to be my TNC?" The answer is, "You can." Bob Richardson, W4UCH, has had packet-radio software for the TRS-80 computers for several years, and it works quite well. There are two reasons that groups like VADCG and TAPR designed and distributed dedicated TNCs. First, a dedicated TNC can be used with any computer that has a serial port, while a program implementing a packet radio node will probably only run on a single type of computer. The second reason for using dedicated TNCs is that they leave the station computer free to run applications programs that use the TNC like a modem.

The "software approach" to packet radio is, however, a valid and useful approach. Many hams who are unwilling to buy TNCs would try packet radio if all they had to do was enter and run a program. So, if you are considering writing packet-radio software for some computer, go to it! Remember, though, that implementing the AX.25 protocol is a major project, and in order to be useful, a program must implement the whole protocol correctly. To avoid problems, be sure to work from a up-to-date specification of AX.25, Version 2.0. This specification is available from the ARRL for $8.00.

We know of packet software projects for the Vic-20 and the Apple 11 computers. If you know of other efforts that are under way, send a note to Gateway.

Ed.

INTERFACE COLLECTION

The following message is forwarded from Lyle Johnson, President of Tucson Amateur Packet Radio Corporation:

"I am compiling a list of hardware and software that can be used by a computer neophyte to interface to a TNC, such as the AEA, GLB, Heathkit, TAPR or VADCG TNCs.

"I am especially interested in what cables are needed, other than the simplest form of RS-232 with DB-25 connectors at each end.

"I would also like to know about any simple, cheap or public domain terminal programs available. The idea is to make it as painless and cheap as possible for a packet newcomer to get on the air.

"It is particularly important that I get information about the following computers:

Commodore Vie-20 and C64
IBM PCjr and PC
Radio Shack TRS80-1,-3,-4 and CoCo
Apple II, II+, //e, //c and Macintosh
TI 99/4 and 99/4A.

"Any listings of simple, dumb terminal programs and sources of cheap commercial programs are also welcome."

From WA7GXD.

COMPUSERVE HAMNET

If your interest in data communications is growing, and you would like to take part in a continuing discussion of issues in packet radio, consider joining HAMNET on CompuServe. If you are already on CompuServe, type "GO HOM 11" to get to HAMNET. If you would like to get on Compuserve, contact a local computer store and ask about it. CompuServe connect fees are reasonable, and the

stations on VHF, he has been doing a lot of operation on HF. There is, of course, talk of linking Charleston and Morgantown. So, if you live on a hilltop in West Virginia, look for **WB8CQV, N8BHI** and **K8LG** on packet radio.

From **WB8CQV.**

ACTIVITY IN NEW YORK

There is a growing packet-radio group in Rochester, New York. The city has about 12 active stations, and 12 more interested parties. Most of the stations are using GLB PK-1 **TNCs.** Rochester is an interesting area from which to pursue **east-***west* linking efforts; with a little effort Rochester could be connected to the Midwest and to **EASTNET,** beginning a transcontinental link.

The Rochester group (which has no formal name) has been recruiting amateurs from the ranks of the local computer clubs. This is a good idea for all packet-radio groups. Give a demonstration of Amateur packet radio to a local computer club; computer hobbyists **may** be surprised to find that amateur radio has something to offer them. As well as helping your packet club grow, these computer-oriented amateurs can be an asset to all of amateur radio.

To contact the Rochester packet radio group, send a letter to:

Fred Cupp, **W2DUC**
27 Crescent Rd.
Fairport, NY 14450.

Prom **W2DUC.**

PACKET RADIO IN PHOENIX

While those of us with TAPR **TNCs** tend to think of Arizona as a **hotspot** of packet activity, local networks have been slow to get started there. Dean Norris, **K7NO,** recently sent **Gateway** a report on activity in the Phoenix, Arizona, metropolitan area. Dean and Wes Morris, **K7PYK** are both active on HF packet. Wes, in Scottsdale, Arizona, is running a WORLL **MailBox** on VHF and **HF,** providing a route for traffic into and out of the area. There are a few other stations that are active only on VHF.

Dean and Wes are interested in optimizing filters and modems for HF packet. Dean has observed that there is a critical range of IF and audio filtering which yields acceptable results under various noise and interference conditions. Wes is looking for packet operators who have experimented with the Yaesu **FT-980** on packet. If you are interested in HF packet radio, listen on 14.100 MHz and 10.147 MHz. All stations are using **300-**baud transmission rate and **200-Hz** shift.

From **K7NO.**

NETWORK NOTES

Here are some other short notes on Amateur packet activity:

Alabama -- The Birmingham packet network is the focal point of packet traffic handling by **Navy-**Marine Corps MARS. This network is being coordinated as part of the Navy-Marine Corps Mars High-Tech program, under the guidance of Jim Gr if ith, **WA5RAX.**

Georgia -- Macon is getting a packet repeater, and the path from Atlanta to Jacksonville (Florida) is getting better every day.

Illinois -- The Chicago Amateur Packet Radio Association **(CAPRA)** has formed a Level-3 Software Development Group. This group is investigating ways to provide high-level linking to the Chicago area. The effort will center around the **9600-**bit/s, **220-MHz** modems developed by CAPRA member Steve Goode, **K9NG.**

Ontario -- There is an AX.25 packet network taking shape in the Hamilton area. Jim **Symes, VE3NBN,** reports that there are already 4 nodes on the air, with a digipeater soon to be operational on 145.01 MHZ.

Via **DRNET, HAMNET.**

FOURTH ARRL NETWORKING CONFERENCE

Here is some more information about the Fourth **ARRL** Amateur Radio Computer Networking Conference, to be held in San Francisco, California, on March 30.

All technical sessions for the conference will be held in room 232. This room is in the East Wing of the Moscone Convention Center. Here is a new schedule of events for the conference:

1030 - Opening remarks and keynote address by Paul Rinaldo, **W4RI** .
1045 - Pete's Packet Primer by Pete Eaton, **WB9FLW.**
1130 - Papers on packet-radio applications.
1200 - Panel session on packet-radio applications by Andy Cromarty, **N6JLJ.**
1230 - Lunch Break.
1330 - Papers on digital communications and packet radio .
1800 - End of technical papers.

If you are a newcomer to packet radio, be sure to attend the Packet Primer. This session should prepare you for the technical papers that will be presented later in the conference.

The voice coordination frequency for the conference is **146.19/.79** MHz, linked to **443.1/448.1** MHz. This repeater, **WB6FDT/R** is being provided courtesy of the Telephone Pioneers.

To whet your appetite for the conference, here are the abstracts for some of the papers that will be delivered:

"Packet Radio Timing Considerations," by David Engle, **KE6ZE** -- This paper presents an analysis of existing packet radio systems and equipment **(2** Meter **AFSK).** Both **dual-** and single- frequency repeater efficiencies are analyzed. The results demonstrate the relative inefficiencies of the existing networks. These inef f ic ienies reduce the effective capacity of a **1200-baud** channel to **300-**500 baud. In order to correct some of these inefficiencies a few suggestions are offered.

Gateway

Vol. 1, No. 14
Feb. 26, 1985

The ARRL Packet-Radio Newsletter

REPEATER BAN LIFTED

On February 21, the FCC released an Order rescinding the repeater moratorium originally declared in Paragraph 10 of the Notice of Proposed Rulemaking, PR Docket 85-22. The Commission stated that "filings by the American Radio Relay League (ARRL) and by the Tri-State Amateur Repeater Council (TSARC) raise serious difficulties with the imposition of the moratorium, including financial hardship to those with repeater construction in process. The problems persuade us to rescind the moratorium. Instead we will seek a permanent solution to all issues relevant to solving questions of interference and congestion by and to stations in repeater operation."

So, packet networks can continue to grow. Repeater coordination and packet networking are still important topics. The FCC requests "substantive comment on all issues relevant to solving questions of interference and congestion by and to stations in repeater operation." This includes packet repeaters. If you or your Network Coordinating Agent (NCA) have opinions on the question of repeater coordination, be sure to bring those opinions to the attention of the FCC.

via W1UED, ED.

CENTRAL NEW ENGLAND COORDINATION

There is some confusion about the proposed Tri-State Packet Radio Council (TSPRC). People have been confusing TSPRC with TSARC, the Tri-State Amateur Repeater Council. TSPRC is not affiliated with TSARC, and at the first meeting of the packet radio council, a name change will be discussed. The name of the packet radio council is similar to that of the repeater coordinating council because the packet radio council intends to cover the same area as TSARC. This area includes Connecticut, Northern New Jersey, Long Island, New York City, and Dutchess and Orange counties in New York. All packet-radio clubs in this area are invited to join the Network Coordinating Council. The first meeting of the council will be in conjunction with the Trenton Computer Fest, April 21, in Trenton, New Jersey.

Ed.

TEXAS FREQUENCY COORDINATION

On February 16, the Texas VHF-FM Society membership passed a motion approving packet radio operations on 145.01, 145.03, 145.05, 145.07, and 145.09, 223.56, 223.58, 223.60 and 441.0 MHz. The Society could not decide whether it should coordinate wide-area digipeaters. The 220.5 to 221.3 MHz section of the 220-MHz band is heavily used in the Dallas area. It is possible, however, that one 100-kHz channel in this section of the band will be coordinated to facilitate out-of-state linking.

From WA5MWD.

440-MHz PACKET

Although not a lot of attention has been paid to packet operation in the 70-cm band, several frequency coordinating bodies have coordinated packet channels on this band. Bill, KB2CQ, reports that he has a GLB PK-1 digipeater on 441.000 MHz in Queens, New York. If you are active on 70-cm packet radio, send a report to Gateway.

In a recent issue of Gateway we incorrectly reported the 70-cm frequency coordinated for packet radio in Northern California. The packet channel is 441.500 MHz.

via KB2CQ, NI6A.

CINCINNATI PACKET CLUB

Cincinnati, Ohio, now has its own packet-radio group. The group, called the Cincinnati Amateur Packet Radio Experimenters Society (CAPERS), was formed after a packet forum at the Cincinnati ARRL convention, February 23. Although there are no packet stations on the air in Cincinnati, several of the group members have TNCs on the way. The group's first task will be to establish a link to the active packet network in Columbus. To contact CAPERS, write:

John Schroer-IV, KA8GRH
984 Halesworth Drive
Forest Park, OH 45240.

Ed.

WEST VIRGINIA PACKET

The packet network in West Virginia is growing. John Jones, WB8CQV, recently sent us a message via the WORLI HF MailBox. John, in Charleston, says that he knows of two other packet stations, both in Morgantown. Although John can't reach these

CORRECTION

In the item **"12-V** DC SUPPLY FOR **TNCS" in Gateway**- number 12, there were two errors. The list of those who developed the power supply should include Don Brown, **NØBRZ,** and the item should be attributed to **NØCCZ,** not **NØCCX.** Sorry about **that.** The power supply was on display in Tucson, and it looked like a simple, useful project.

Ed.

PACKET AT HAMFESTS

There will be packet-radio booths or forums at the following upcoming hamfests:

Feb. 23 -- Midwinter Technology Fair, Fridley, Minnesota. Talk-in frequency **147.ØØ/.6Ø** MHz.

Feb. **23** -- ARRL Ohio State Convention, Cincinnati, Ohio. Packet forum at **1:ØØ** PM.

REPRODUCTION OF GATEWAY MATERIAL

Material may be exerpted from Gateway without prior permission, provided that the original contributor is credited and Gateway is identified as the source.

Gateway: The ARRL Packet Radio Newsletter is published by the

American Radio Relay League
225 Main Street
Newington, CT 06111

Jeffrey W. Ward, **K8KA**
Editor

Larry E. Price, **W4RA**
President

David Sumner, **K1ZZ**
General Manager

Subscriptions are available to ARRL members and nonmembers, at the following rates:

For 25 issues of Gateway:
- In the U.S.
  - ARRL Member $6 .OO
  - Nonmember $9 **.00**
- In U.S., Canada, and Mexico by First Class Mail
  - ARRL Member $11.00
  - Nonmember $14.00
- Elsewhere by Airmail
  - ARRL Member **$14.00**
  - Nonmember $17.00.

Prices are subject to change without notice. Remittance must be in U.S. funds, and checks must be drawn on a bank in the U.S.

Gateway: The ARRL
Packet Radio Newsletter
American Radio Relay League
225 Main St.
Newington, CT 06111

Nonprofit Organization
U.S. Postage
PAID
Hartford, **Conn.**
Permit No. 2929

SOUTHERN WESTNET COORDINATION

The packet-radio community in Southern California has formed the Southern California Digital Communications Council (SCDCC). The purpose of this council is similar to that of groups previously formed in Northern California and the Mid-Atlantic States: to provide a single point of contact to the area frequency coordinating bodies and to serve as a clearing house for information and a forum for discussion of effective use of network resources. The first act of SCDCC was to petition the local two-meter coordinating body for allocation of 145.01 MHz as an inter-region packet linking frequency.

Membership in SCDCC is open to anyone with an interest in digital communication. All California packet-radio operators from Santa Barbara to the Mexican border are encouraged to join. The SCDCC mailing address is:

SCDCC
P.O. Box 6Ø26
Mission Hills, CA 91345.

Via NK6K.

FREQUENCIES COORDINATED

The New England Spectrum Management Committee met on January 26, 1985 and made the following frequency allocations for packet radio:

2 meters: 145.01 to 145.09 MHz, for operation with less than 5 kHz deviation.

1 1/4 meters: 5 1ØØ-kHz channel6 between 220.5 and 221 MHz and 1Ø 2Ø-kHz channel6 from 221.ØØ to 221.18 MHz.

70 cm.: 1Ø 1ØØ-kHz channels from 43Ø to 431 MHz and a single frequency at 441.00 MHz for 5-kHz deviation operation.

This allocation scheme follows a plan discussed at the Third ARRL Amateur Radio Computer Networking Conference and published recently in the newsletter 220 NOTES.

In Florida, the Florida Repeater Council has coordinated 145.01 MHz as the state-wide packet-radio frequency. In add it ion, they have coordinated 221.72, 221.78 and 221.40 MHz as 2Ø-kHz channel6 and 22Ø.57 for 1ØØ-kHz bandwidth operation,

Via K4NTA and AG1F.

NTS AND PACKET RADIO

Don Haney, KA1T, is running a packet-radio bulletin board system (PBBS) dedicated to NTS traffic. Don'6 PBBS guides users who are unfamiliar with NTS radiogram format through the message originat ion process. The PBBS is part of the growing network of WØRLI MailBoxes that are forwarding traffic around New England on 145.01 MHz. Don is looking for stations on HF to extend this network.

In the January issue of the NEPRA Packetear, Don writes that 110 pieces of formal NTS traffic went through his PBBS in December. Forty-five percent of that traffic was for delivery outside New-England and 35% was between the Eastern and Western Massachusetts. The balance of the traffic was for delivery in New Hampshire, Connecticut and Eastern Massachusetts.

If packet traffic is originated on a crowded channel or entered by keyboard, each message takes about 3 to 5 minutes -- longer than it would take to originate the same message on FM or CW. The advantages of packet radio are realized when the messages are being forwarded from computer to computer automatically. With automatic forwarding, messages can be passed at times that would otherwise be inconvenient, and they can be passed without introduction of errors.

VIA PACKETEAR.

MODEMS AVAILABLE

Jerry Quimby, N4AJH, is making available a limited number of PC boards for EXAR modems. Jerry is selling a finished PC board, parts placement diagram, parts list, schematic diagram and tune-up instructions. The PC board measures 1 1/2 by 3 in., and can be used for either 3ØØ-bit/s or 12ØØ-bit/s AFSK operation. If you are interested, send $8.50 in check or money order to:

Jerry Quimby, N4AJH
2677 Hereford Rd.
Melbourne, FL 32935.

These modems are already in service on some heavily-used Florida digipeaters, and the chips in the circuit are the same ones used in all of the available TNCs. Even though you must supply your own parts for the PC board, this is an inexpensive modem.

VIA DRNET.

TWO-PORT DIGIPEATER

If you want to link two packet networks that are on different frequencies, you may have to use complex digital and audio contraptions. Most of the available schemes involve auxiliary links between digipeaters on different frequencies, or single digipeaters that use audio mixing to communicate simultaneously on two channels. While these solutions to the frequency-translation problem work, they are not always the simplest or most efficient solutions.

A simple solution to this problem is a digipeater with two independent packet I/O ports and enough internal logic to route packets from one port to another. Jon Bloom, KE3Z, has written a program that implements such a digipeater. The software runs on a Xerox 82Ø, using the serial I/O port (SIO) to send and receive packets. It has several mechanisms for routing packets from one port to another. The program resides in ROM, and is suitable for remote digipeaters.

If you are interested in this software, send an 8" SSSD disk to Gateway. In return, you will receive the program and appropriate documentation.

Ed.

On February 2nd, packet-radio operators from Northern New Jersey, Eastern New York and Connecticut met to discuss the formation of a Network Coordinating Agent (NCA). Those present agreed to form a NCA called the Tri-State Packet-Radio Counc il (TSPRC). TSPRC will be a council of packet radio clubs from the areas covered by the Tri-State Amateur Repeater Council (TSARC), a frequency coordinating body. All clubs within TSARC jurisdiction are invited to attend the init ial meeting of TSPRC, at the Trenton Computer Fest, April 21st. For further information on TSPRC, contact:

TSPRC, c/o Jeff Ward, K8KA
52 Alden St., Apt. 2Ø2
Hartford, CT 06114.

Ed.

HEATHKIT TNC

Those lucky enough to attend either the Miami Eamfest or the TAPR Annual Meeting have seen the second entry into the "TAPR-clone" field. The Heathkit HD-4Ø4Ø is a TAPR TNC in a brown and beige low-profile cabinet. Status-indicating LEDs are grouped on the left side of the front-panel, and the only switch on the unit is an ON/OFF switch on the right side of the panel. Inside, the HD-4Ø4Ø is purely a TAPR TNC, complete with modem-configuration headers and AC power supply. The kit will sell for $299.99 and should be available by the second week in April.

The kit will feature an “improved” TAPR manual. Considering that the TAPR manual is one of the best seen in amateur radio, the Heathkit manual should be a superb document. Parts for the HD-4Ø4Ø will come on an adhesive strip, in the order that they are required in kit construction. This packaging technique makes parts-list checking unnecessary, and should make construction go quite quickly.

The HD-4Ø4Ø will bring two new marketing techniques to Heathkit: For the first time, the Heathkit TNC manual will explain how to connect a Heathkit accessory to non-Heathkit radios. Also, Heathkit will probably be selling the HD-4Ø4Ø in a specially-priced package with another manufacturer’s assembled and tested amateur radio transceivers.

Wayne Wilson, Heathkit’s Product Line Manager for General Consumer Products, attended the TAPR meeting. Mr. Wilson believes that packet radio could breath life into the sagging amateur-radio industry. Heathkit is eager to build connections between amateur radio and computers, hoping that technically-inclined youth can be drawn back into amateur radio.

TAPR will continue to produce the TNC kit as long as there is demand for them. Most of TAPR's research and development funding comes from TNC sales. TAPR made the TNC available to commercial interests believing that packet radio is a field with room for more than one manufacturer. With nearly 2ØØØ TAPR TNCs sold, they are being proven right.

Ed.

FOURTH COMPUTER NETWORKING CONFERENCE

The Fourth ARRL Computer Networking Conference will be held in San Francisco on March 3Øth, 1985. The conference, co-sponsored by the ARRL and the Pacific Packet Radio Society, will be held in conjunction with the Tenth West Coast computer Faire, which runs from March 3Øth through April 2nd at the Moscone Convention Center. The tremendous growth and interest in packet radio terminals, equipment, networks and applications promises to make this conference one of the largest and best-attended ever.

The conference schedule for Saturday, March 30th follows:

1000 - Introduction to Packet Radio
11ØØ - Panel session on Applications of Packet Rad io
12ØØ - Lunch Break
13ØØ - Technical papers delivered
18ØØ - End of technical session
2ØØØ - Conference dinner (price and location to be announced later).

Through special arrangements with the Faire and the ARRL, a special pre-registration fee of $2Ø will purchase a four day ticket to the Faire and the ARRL conference proceedings. Send a check for $2Ø and a s.a.s.e. to:

Hank Magnuski, KA6M
311 Stanford Avenue
Menlo Park, CA 94Ø25.

The deadline for submitting papers for the conference proceedings is March lst, 1985. If you need an author’6 kit, contact the ARRL.

Via KA6M.

WESTNET LINKED

The first reliable, 24-hour link between Northern California and Southern California was completed on February 3. In an amazing burst of energy, several digipeaters in Northern California were brought up in the last few weeks, extending the network 2ØØ miles south. Coincidentally, a critical mountain range near Santa Barbara had just been bridged by amateurs from Southern and Central California. Until 145.01 MHz is allocated in Southern California, an audio link on 45Ø MHz connects the 145.Ø1-MHz digipeater at Arroyo Grande with the 145.35-MHz machine at Santa Barbara. Eventual removal of this audio link will make the Los Angeles to San Francisco path a three-digipeater connection. Now, stations from San Francisco can use a six-digipeater path to connect with Los Angeles. For packet operators in California, the completion of this path is a milestone and an indication of better things to come.

For those keeping track of packet-radio DX records, the link between Northern and Southern California is approximately 53Ø miles long, and will be extended north another 1ØØ miles soon.

Via NK6K, KA6M.

Gateway

Vol. 1, No. 13
Feb. 12, 1985

The ARRL Packet-Radio **Newsletter**

TAPR MEETING

I have just returned from the Annual Meeting of the Tucson Amateur Packet Radio Corp. (TAPR). Most of the "hotbeds" of U.S. amateur packet radio activity were represented at this day-long forum in Tucson. It was an exciting meeting: Several maps showing packet activity throughout the U.S. illustrated that packet radio is indeed growing rapidly. Last year, most people were worrying about stimulating packet activity in isolated areas; this year, the emphasis was on linking these isolated areas into a network. There were representatives from AEA and Heathkit, two of the commercial manufacturers who have "joined the packet-radio revo lut ion." Perhaps the most exciting items demonstrated at the meeting were the 9.6-kbit/s modem6 designed by Steve Goode, K9NG. With these modems transmitting data eight times as fast as current modems, a truly multi-level network can be built. (In such a network, a high-speed inter-city link would handle long-haul traffic for several 1200-bit/s conversations.) That TAPR has supported and organized these projects proves that the group responsible for the most popular TNC is still helping advance the amateur-radio state of the art.

One of the purposes of the amateur radio service, as stated in Part 97 of the FCC rules and regulation6 is to "contribute to the advancement of the radio art." Unfortunately, for many years, commercial technology has been advancing more quickly than amateur radio. Presentations at the TAPR meeting, made it clear that amateur packet radio is advancing the state of the radio art. Both the U.S. Forest Service and Army MARS sent representatives to the meeting to find out how they could use amateur-generated packet-radio techno logy. Several speakers mentioned that their local Red Cross emergency operation center6 are equipped for packet radio. In the west, where search-and-rescue efforts must cover vast areas of wilderness, the Civil Air Patrol has begun to use packet-radio for search coordination. TAPR TNCs have even found their way onto military hurricane-investigation airplanes. Amateur6 are once again contributing to the state of the art.

The TAPR meeting reinforced my belief that packet radio is bringing out the best in amateur radio. The technological advances demonstrated at the meeting and the interest in packet radio shown by Heathkit and AEA lead those present to conclude that packet radio will enjoy rapid growth in the month6 and years ahead.

Ed.

PACKET AND THE FCC

In late January, the FCC released Notice of Proposed Rulemaking (NPRM) 85-22, proposing to add definitions of "coordinated repeater," "frequency coordinator ," and "harmful interference" to Section 97.83 of the FCC rules and regulations. The NPRM also proposes adding text to Section 97.85 concerning resolution of conflicts between repeaters. Complete text of this NPRM is available from the ARRL for an s.a.s.e. with $0.54 postage.

In a move that worried many packet-radio experimenters, the FCC stated that "during the pendency of this proceeding there will be a moratorium on new repeater operation in Central Metropolitan Statistical Area6 and Metropolitan Statistical Areas." These areas include many growing packet-radio networks. Packet-radio operators are asking whether this moratorium extends to simplex digipeater operation on previously established or coordinated frequencies. Informally, FCC officials have stated that packet radio is not covered by this repeater ban. While no formal determination has been made, it is clear that the FCC did not consider digipeaters before placing the 9-month ban on new repeater installations.

On February 7, the ARRL filed a Petition for Partial Reconsideration, asking the FCC to drop the moratorium on new repeaters from NPRM 85-22. The ARRL sees both practical and legal problem6 with the ban. Part of the Petition for Partial Reconsiderat ion reads: "the suggested moratorium comes at a very bad time in the evolution of packet-switching networks in the Amateur Radio Service which are dependent on repeaters for their operation. The suggested moratorium will place a chill on the development of packet repeaters and hence the communications experimentat ion which is presently at the forefront of amateur technology." The FCC should consider this request soon.

Comments on NPRM 85-22 must be filed by July 1, 1985. Since the proposed rule changes would have some impact on frequency coordination for packet radio, packet operators should be sure to comment on the NPRM.

Ed.

NETWORK COORDINATION

The recent FCC actions concerning repeater coordination underscore the need for packet-radio operator6 to work with local (and perhaps national) frequency-coordinating bodies.)

PACKET IN THE PRESS

The February issue of The Institute, a publication of the Institute of Electrical and Electronics Engineers, contains an article titled "Hams Exploiting Packet Switching, Satellites, and More." The article discusses the AX.25 standard, the uses of computers in amateur radio, and the growth of amateur packet radio. The author, Glenn Zorpette, interviewed both Paul Rinaldo, **W4RI,** and Jan King, **W3GEY.** Jan and Paul provided plenty of information to show the readers of The Institute that radio amateurs are still part **of the** state of the art.

Ed.

SEND IT!? YOUR PACKET NEWS

When something happens in your packet-radio network, or you have an opinion on a Gateway item, send a note to Gateway. Information is always worth more if it is made available to others.

Ed.

REPRODUCTION OF GATEWAY MATERIAL

Material may be exerpted from Gateway without prior permission, provided that the original contributor is credited and Gateway is identified as the source.

Gateway: The ARRL Packet Radio Newsletter is published **by the**

American Radio Relay League
225 Main Street
Newington, CT 06111

Jeffrey W. Ward, **K8KA**
Editor

Larry **E. Price, W4RA**
President

David Sumner, **K1ZZ**
General Manager

Subscriptions are available to ARRL members and nonmembers, at the following rates:

For 25 issues of Gateway:
- In the U.S.
  - ARRL Member $6 .OO
  - Nonmember $9 .00
- In U.S., Canada, and Mexico by First Class Mail
  - ARRL Member $11.00
  - Nonmember $14 .OO
- Elsewhere by Airmail
  - ARRL Member $14.00
  - Nonmember $17.00.

Prices are subject to change without notice. Remittance must be in U.S. funds, and checks must be drawn on a bank in the U.S.

Gateway: The ARRL
Packet Radio Newsletter
American Radio Relay League
225 Main **St.**
Newington, CT 06111

Nonprofit Organization
U.S. Postage
PAID
Hartford, Conn.
Permit No. 2929

the sunspots will be against us for at least the next three winters, and ve will need some mechanism for delivering the short-haul traffic.

"We hope to use packet radio on VHF to deliver all of the traffic that we cannot deliver on HF. This should be accomplished using packet both within the Sixth Region and between the Sixth Region and other Regions. Since Oregon could provide liaison with the Seventh Region, ve would just need a link into the Twelfth Region (Arizona) to link all of the Pacific Area. A packet link into Arizona might be accomplished through Southern California.

"We hope that, eventually, all NTS Section nets will establish liaison with a packet-radio network. On each packet-radio network there could be a "Section Net Mailbox" for outgoing and incoming messages. Some station would take the responsibility of passing traffic to and from the Section net everyday.

"Eventually there will be enough satellite and HF links to permit each Section to send its traffic directly to the destination Section. In the meantime, we must use whatever links and routing mechanisms we have.

"We would like to hear from any packet operators interested in establishing NTS mailboxes for their section. We are attempting to develop some software to facilitate NTS traffic handling. I have over 25 years experience with NTS, and I'm enthusiastic about the use of packet radio to provide efficient, communications."

Write to:

Don Simon, NI6A
STM EASTBAY
2327 Alva Ave.
El Cerrito, CA 94530

From NI6A.

TCC SKEDS

Don Simon, NI6A, is looking for traffic operators to try making packet TCC schedules on OSCAR 10. TCC is the NTS Transcontinental Corps, who handle long-distance NTS traffic. Don can operate 1200-bit/s, 1000-Hz shift, or 300-bit/s, 200-Hz shift. Interested TCC directors should contact Pat Berry, KN7B, TCC director for the Pacific Evening NTS.

Via NI6A.

UTAH PACKET-RADIO CLUB

The Utah Packet Radio Association (UPRA) was formed on January 5, 1985, in Salt Lake City, Utah.

UPRA was formed to meet several needs of the growing Packet-Radio Community in Utah, the most important of which is the development of packet radio in Utah. Other UPRA projects include the creation of a packet network between Salt Lake City and Denver and the development of links toward the Pacific Northwest and Southern California.

Twenty interested hams, four of whom are already on packet, attended the first UPRA meeting. One of the charter members of UPRA is Ron Todd, K3FR, the ARRL Utah Section Manager.

For further information on UPRA, contact:

David J. Pedersen, N7BHC
President, Utah Packet Radio Assn.
4382 Cherryview Drive
West Valley City, UT 84120

From UPRA.

PACKET RADIO FOR THE IBM PC

Jack Botner, VE3LNY, has developed a peripheral card for the IBM PC that uses an 8273 high level data link controller (HDLC) integrated circuit to provide packet-radio I/O for the PC, The card allows the PC to access the HDLC chip through polling, interrupts or DMA. If you are an experimenter and you own an IBM PC, you will be interested in this project. Complete details, including a circuit diagram, appear in the January issue of QEX, the ARRL experimenter's exchange. Although the article does not cover software to operate the controller, the AX.25 software written by Phil Karn, KA9Q, could be modified to run on the PC.

From QEX.

12-V DC SUPPLY FOR TNCS

Link Haymaker, K0ZCO, and Tim Groat, KR0U, from Colorado, have adapted an inexpensive ($4.95) surplus power supply board for use with the VADCG and TAPR TNCs. The supply is made by Iriichi Tsushin Kogyo Ltd., and is available from several sources, including Radio Shack and BNF Enterprises. The modifications to the supply will convert the negative 5-V output to negative 12-V (as required by the TNCs), and will allow the supply to operate with from 10 V to 15 V dc input. This allows the entire TNC to run from a 12-V battery, permitting portable and emergency operation,

Some changes to the TAPR TNC are required, particularly on older pc boards. VADCG boards work without modification, since they were designed to use external power supplies.

For a copy of the schematic and instructions for this conversion, send a business-sized s.a.s.e. with postage for 1 ounce to:

Rocky Mountain Packet Radio Association
Power Supply Experiments
3775 W 115th Ave.
Thornton, CO 80233.

A power supply and a modified TAPR TNC will be available for inspection and demonstration at the TAPR Annual meeting.

From N0CCX.

of the DCE proved to Conference attendees that low-cost satellites can provide reliable, worldwide, digital communication. AMSAT, VITA, and INTER-PARES hope that this presentation will create interest in and support for the PACSAT project.

Via UoS, NK6K.

PACKET RADIO IN THE ARRL HANDBOOK

Packet-radio operators and experimenters, and those interested in digital communications will find a wealth of new information in the 1985 ARRL Handbook for the Radio Amateur. Extensive editorial additions and revisions, coupled with several new projects have made the Handbook a state-of-the-art publication.

For those who are new to computers and digital techniques, there is Chapter 8, "Digital Basics." This chapter introduces digital logic, digital interfacing, and digital computers, among other topics. If you are looking for a guide to digital electronics, try "Digital Basics."

The "Digital Communications" chapter contains 30 pages of up-to-date information on digital techniques from Morse code to packet radio. Topics range from the RS-232-C interface standard to the OSI packet-switched network model. The chapter ends with a glossary of digital communications terms.

Two of the projects in the "Digital Equipment" chapter should convince you to plug in your soldering iron. For troubleshooting serial interfaces, there is an RS-232-C breakout box. This piece of test equipment can save a lot of frustration when you are trying to hook a TNC to a computer or terminal. The other interesting project is a modem capable of using both Bell-202 and Bell-103 tones. The modem has been used extensively on both HF and VHF packet.

So, take a look at the 1985 ARRL Handbook for the Radio Amateur. For $15.00, you get 1024 pages of useful radio and electronics information.

Ed.

TAPR MEETING

Tucson Amateur Packet Radio Corp. (TAPR), will hold its annual meeting on Saturday, Fe-bruary 9, in the Granada Royale Hometel, Tucson, Arizona. If you can make it to Tucson, even if you are not a TAPR member, come to the meeting. This is a great opportunity to meet packet-radio operators from around the country, hear about growing packet-radio networks, and discuss the issues that face the packet community.

Speakers at the meeting will include Steve Goode, K9NG, and Harold Price, NK6K. Steve will describe his 9600-bit/s, 220-MHz radios. His talk will end with a demonstration of high-speed packet radio-, Harold will talk about the successful UO-11 data communications experiment (DCE) test that were carried out in the end of January. Harold will probably touch on the status of PACSAT, too. TAPR is still lining up other speakers; expect to see many of the leaders in packet radio at the meeting.

If you need further incentive to attend, there will be a free "packet pizza party" on Friday to kick off the meeting.

The talk-in frequency will be 146.28/.88 MHz.

Via WB9FLW.

MIAMI PACKET FORUM

There will be a Packet Radio Applications Forum at the Miami Hamcation on Sunday, February 3. This forum will feature Lyle Johnson, WA7GXD, Pete Eaton, WB9FLW, and Paul Rinaldo, W4RI. These nationally recognized packet-radio experimenters will try to provide some answers to the question "what do we do with packet radio now that it's here?" The forum is sponsored by the Florida Amateur Digital Communications Association (FADCA). For further information, contact Joel Kande 1 at (305)-596-9373.

Via DR NET.

PACKET RADIO DX CONTEST

To prove that packet radio is "not just for computer hackers," the Florida Amateur Digital Communications Association (FADCA) is sponsoring a packet-radio DX contest. All contacts must be made above 144 MHz. There is no limit to the number of digipeaters that can be used, but they must all be above 144 MHz. Satellite contacts are not allowed. Contacts must be in "real time;" that is, no store-and-forward mailboxes may be used. Distance will be measured between the endpoints of the contact, not along the route taken by the packets. A valid contact is the acknowledged exchange of name and QTH, in the connected mode. Contacts made between December 31, 1984 and March 31, 1985 will count. All entries must be received by April 15. Send them to:

Ted Huf, K4NTA
1829 Pinetree Way
Stuart, FL 33949.

Have fun!

From K4NTA.

NTS TRAFFIC ON PACKET

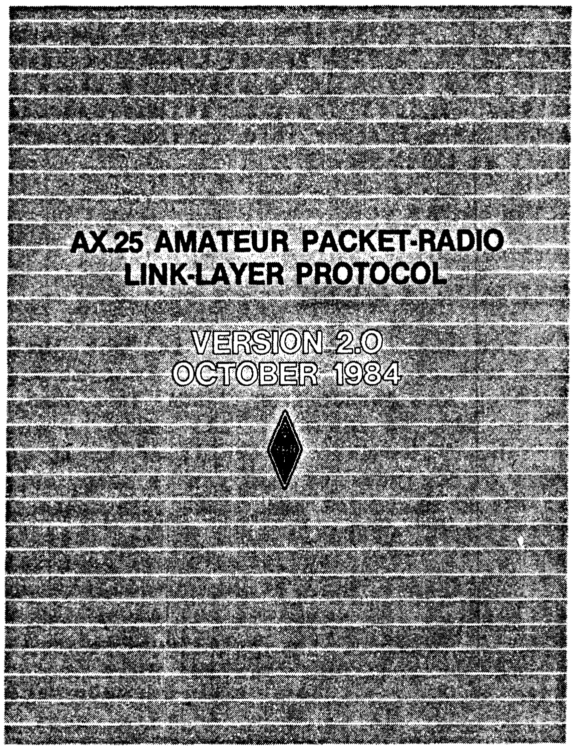
Gateway received the following letter from Don Simon, NI6A, East Bay Section Traffic Manager (STM):

"While we are in the null of the sunspot cycle, band conditions on 80-meter traffic nets will continue to get worse. Many 'dyed-in-the-wool' traffic handlers are attempting to use packet radio to help NTS move traffic around the country. Here on the Pacific West Coast, we hope to have packet links established north to the Oregon border and south to the Southern California Section Net before the next holiday-rush season. It may sound too early to plan for next winter, but to establish a reliable NTS packet network will take quite a bit of organization. Remember,

Gateway

Vol. 1, No. 12
Jan. 29, 1985

The ARRL Packet-Radio **Newsletter**



AX.25 AMATEUR PACKET-RADIO
LINK-LAYER PROTOCOL

VERSION 2.0
OCTOBER 1984

PROTOCOL SPECIFICATION AVAILABLE

In October 1984, the ARRL Board of Directors approved the AX.25 link-layer protocol presented to them by the ARRL Ad Hoc Committee On Amateur Radio Digital Communications. This protocol is the familiar AX.25, with a few small changes that were requested by packet-radio experimenters. Now, the protocol specification for AX.25, version 2.0, is available from the ARRL. This is a formal protocol specification, providing all the informat ion necessary to implement AX.25.

If you are building a TNC, of if you are simply interested in how your TNC works, you should get a copy of "AX.25 Amateur Packet-Radio Link-Layer Protocol." The document is available from the ARRL for $8.00 in the U.S. and $9.06 in Canada and elsewhere.

Do not be alarmed that this document is subtitled "Vers ion 2.0." The version number distinguishes this, the first official specification of AX.25, from previously available specifications. Version 2.0 of AX.25 adopts the Poll/Final procedure proposed by Phil Karn, KA9Q. This procedure is compatible with existing implementat ions of AX.25.

The availability of a protocol specification for AK.25 opens the way for widespread standardization of amateur packet radio. Such standardization makes it possible for manufacturers to enter packet radio with confidence. It also makes international projects like JAS-1 and PACSAT more likely to be compatible with available TNCs.

Congratulations to all of the experimenters, implementors, and operators who have made AK.25 a robust , mature protocol.

Ed.

DCE SUCCESS

Amateur radio operators affiliated with AMSAT, VITA, INTER-PARES, and the University of Surrey have successfully used the data communications experiment (DCE) aboard UoSAT-OSCAR-11 to exchange messages between Los Angeles, Hawaii, and the UK. On January 16, at the Pacific Telecommunications Conference in Honolulu, a portable station assembled by WH6AMX, WA3ZIA, and VE3FLL received messages that had been previously stored in the DCE by Harold Price, NK6K (in Los Angeles) and the UoSAT command station (in Surrey, England).

The DCE was included on UO-11 to provide a "proof-of-concept" experiment for worldwide, store-and-forward packet communications. Designed and built by AMSAT and Volunteers In Technical Assistance (VITA) teams from the U.S. and Canada, the DCE will evaluate the hardware, software and communications protocols that will be required for a fully operational packet satellite -- PACSAT.

The DCE has been providing a "digital bypass" necessary to overcome communications problems that developed just after the launch of UO-11. In early January, G8NEF developed software that enabled the spacecraft's primary computer to manage the "bypass," making the DCE available for experiments.

A Canadian group (headed by Larry Kayser, WA3ZIA/VE3), in conjunction with NK6K, developed a simple store-and-forward program for the DCE. The software was loaded into UO-11 by NK6K, and the experiments were under way.

WA3ZIA and Lionel Pett, VE3FLL, attended the Pac if ic Telecommunications Conference to discuss the potential uses of PACSAT. They enlisted Rick Dittmer, WH6AMX to help them assemble and operate a portable ground station at the Conference. Despite poor weather, the team was able to receive messages from the DCE and copy the UO-9 "Special Events Bulletin." This successful demonstration

SOFTNET WORKSHOP

The SOFTNET User's Group (SUG), in cooperation with Linkoping University, will hold its third SOFTNET Workshop on Sunday May 11, 1985, at Linkoping University, Linkoping, Sweden. Technical papers are solicited on high-speed packet-radio hardware and propagation, distributed routing in packet-radio networks, SOFTNET applications, and SOFTNET specification issues. Camera-ready summaries consisting of less than two standard A4 pages should be received by SUG by MAY 1st. Proceedings from the workshop will be available through SUG. For more information contact:

Per Lundgrcn
SUG, Department of Electrical Engineering
Linkoping University
S-581 83, Linkoping
SWEDEN.

From Softaet News.

REPRODUCTION OF GATEWAY MATERIAL

Material may be exerpted from Gateway without prior permission, provided that the original contributor is credited and Gateway is identified as the source.

Gateway: The ARRL Packet Radio Newsletter. is published by the

American Radio Relay League
225 Main Street
Newington, CT 06111

Jeffrey W. Ward, K8KA
Editor

Larry E. Price, W4RA
President

David Sumner, K1ZZ
General Manager

Subscriptions are available to ARRL members and nonmembers, at the following rates:

For 25 issues of Gateway:
- In the U.S.
  - ARRL Member $6.00
  - Nonmember $9.00
- In U.S., Canada, and Mexico by First Class Mail
  - ARRL Member $11.00
  - Nonmember $14.OO
- Elsewhere by Airmail
  - ARRL Member $14.00
  - Nonmember $17.00.

Prices are subject to change without notice. Remittance must be in U.S. funds, and checks must be drawn on a bank in the U.S.

Gateway: The ARRL
Packet Radio Newsletter
American Radio Relay League
225 Main St.
Newington, CT 06111

Nonprofit Organization
U.S. Postage
PAID
Hartford, Conn.
Permit No. 2929

with the higher and lower echelons of NTS.

"Points 3 and 4 are the real stumbling blocks for a packet net. Regular interfacing to NTS is a chore! Stations must be available at prescribed times on a reliable, regular basis in order for the traffic flow to be effected properly.

"None of the above is to insinuate that a packet network cannot become officially affiliated with NTS. I just wanted to point out that a monthly net report is NOT the sole criterion for NTS affiliation."

From WA4STO.

DIGITAL **COMMUNICATIONS** BOOK

Gateway received the following book review from Pete Eaton, WB9FLW:

"While browsing through the bargin table at the local Radio Shack, I happened to glance at their rack of tutorial books. To my suprise, one of the titles was Understanding Data Communications. For the last several years, newcomers to packet radio have been asking for some introductory reference material written in plain language. Understanding Data Communications, developed and published by the Texas Instruments Learning Center, comes close to filling the bill.

"Each chapter introduces you to a fundamental concept in digital communications. Topics covered include an overview of data communications, data terminals, types of messages and transmission channels, asynchronous and synchronous operation, and satellite communications.

"Of special interest to amateur packet operators are chapters covering 'Protocols and Error Control,' 'Local Area Networks,' and 'Network Design and Management.' A special treat is Chapter 9, 'Architecture and Packet Networks,' which uses X.25 as an example of a packet-switching network protocol.

"The whole book is written in clear, down-to-earth language. Once a novice has read all the chapters, I'm sure that he will better understand and appreciate the potential of Packet Radio in the Amateur Radio Service. If you are getting confused by all the talk of 'protocols,' 'LANs,' 'FSK,' 'PSK,' and 'level 3,' give yourself a present from Radio Shack."

[Remember that a thorough introduction to packet radio and several projects of interest to packet operators appear in the 1985 edition of The ARRL Handbook for the Radio Amateur. -- Ed.]

From **WB9FLW.**

PACKET RADIO IN JAPAN

Using a TAPR TNC as a debugging aid and compatibility tester, several Japanese amateurs are implementing their own TNC. The TNC will use a Z80 CPU and an 8273 HDLC chip. JA1MIR reports that JI1BXM is writing the AX.25 software for the TNC and that it should be complete by the end of January.

The Japanese group will be building an OSCAR 10 station to communicate with U.S. 'packet experimenters. They are particularly interested in internetwork (level-3) protocol development and high-speed modem research. Look for them on the satellite.

Takemi, JA1MIR, notes: "Designing our own TNC through understanding the AX.25 specification is great fun. I haven't been involved in such an exciting amateur-radio project for more than 10 years!"

Via **NK6K.**

PACKET RADIO IN ENGLAND

The following report on packet-radio activity in England was sent to us by Reg Brake, G8QR:

"In Norwich, there are now five stations equipped with TAPR TNCs. Those stations are G3LDI, G3PMQ, G4RSP, G6JTH, and myself. In addition, there are about fifteen local stations equipped with the software packet-radio program that runs on the BBC microcomputer. This program was developed by G6GIX and G8WJL at the Cambridge University Computer Laboratory.

"All of the stations equipped with TAPR boards have worked successfully on VHF. G3PMQ and I have also made contact with Ipswich. This activity is on 144.68 MHz, using 1200 bauds, 1000-Hz shift. In the last month, however, we have modified the TAPR boards to switch from wide shift to narrow shift at the touch of a button. The physical arrangement for this was designed by G6JTH and it functions very well. Since that time, we have been doing some HF packet operation.

"On 14.103 MHz, we made the first England-to-Germany packet-radio QSO, with DL2MDE. As I speak German, I initiated the contact, but both G3PMQ and I maintained an hour-long QSO with solid copy both ways. We have heard the American packet station [W0RLI] at 14.080 MHz, but signals were weak at the time, and we were not equipped with a tuning indicator. Since then, two stations have built the TAPR LED tuning indicator and use this very successfully on HF. We are looking for contacts on 14 MHz, although we are uncertain of the preferred frequencies. None of us is yet on OSCAR 10, so we are anxious for contacts on HF."

The winter copy of DATACOM, the journal of the British Amateur Radio Teleprinter Group (BARTG) contains a packet-radio column by Ian Wade G8NRW. In that column, Ian provides a list of about 60 stations active on packet radio in England.

If you are going to be in England this spring, note the following (also from BARTG DATACOM): At the RSGB VHF Convention at Sandown Park on March 23, 1985, Ian Wade, of BARTG, is presenting a talk entitled "How Packet Radio Works." For more information contact

Ian Wade, G8NRW
7 Daubeney Close
Harlington, DUNSTABLE
Bedfordshire, LU5 6NF
ENGLAND.

From G8QR and **DATACOM.**

discussion of how present TNCs handle retries and of a couple of bugs in the TAPR TNC-1 software (version 3.3) that add to network congestion.

There are now three UNIX nodes known to be operating on EASTNET. They are run by Phil Karn, Jim Kutsch, KY2D, and Brian Lloyd. Brian gave details on how to get USENET mail onto EASTNET through these systems.

Via **HAMNET.**

SOUTHNET XI UPDATE

We have received further details on the SOUTHNET II conference to be held at Georgia Tech, on November 23 and 24. The featured presentations are:

- Lyle Johnson, WA7GXD, and Pete Eaton, WB9FLW, on the TAPR Network Node Controller.

- Ed Jackson, WB2OIF, from GLB, will discuss the AX.25 protocol, the PK-1 command set and future GLB projects.

- Demonstrations of all available packet hardware.

- SOUTHNET organizational meeting.

- SOUTHNET awards, including the "SOUTHNET Packeteer of the Year."

Technical sessions will include a report on the progress of digipeaters in the SOUTHNET region, an update on the 9600-baud modem, a status report on 220-MHz linking, discussions of various networking proposals, sessions covering the WØRLI Gateway and MailBox, and a roundup of Xerox 820 information. There will also be an introductory presentation for those just getting started or interested in packet radio. The conference will be capped by a banquet on Saturday evening.

Members of GRAPES, the Georgia Radio Amateur Packet Enthusiasts Society, will provide transportation to and from the airport. Those that need this service should contact Bill Crews, WB2CPV, at least 3 weeks prior to the conference. With super-saver air fares available, you should prepare early to attend SOUTHNET II.

There will be a final mailing of conference information in early November, including a complete agenda, maps of the meeting location and information on sightseeing in Atlanta. To get this information, send an SASE to

Bill Crews, WB2CPV
1421 Hampton Ridge Road
Norcross, GA 30093.

Via **DRNET.**

CENTER FOR WØRLI ROUTING INFORMATION?

With new WØRLI MailBox and Gateway stations coming on the air frequently, it is hard for sysops (system operators) to keep track of all of the interconnected PBBSs. In order for the system to work, each PBBS must know which of its neighbors to send mail to for every PBBS in the network.

While it is not practical to keep track of all PBBS users and their whereabouts, it is a smaller task to keep track of each PBBS and the paths through which it can be accessed. David Dodell, WB7TPY, has suggested that someone maintain a database of MailBoxes and Gateways, allowing sysops to get timely information about new systems and paths. If you have any comments about this, send them via packet to WB7TPY @ K7PYK or put them in the mail to Gateway.

Via **HAMNET.**

SOUTH TEXAS PACKET

The Texas Tech University Amateur Radio Club will be putting up several new repeaters this fall. One of the machines will be a packet-radio digipeater on 145.01 MHz. Ronald Cole, N5HYH, is in charge of the club's station operations, and he is planning to put a packet-radio bulletin board system (PBBS) on the air to stir up some local interest in packet. The other repeaters to be installed by the Texas Tech club include a duplex repeater dedicated to experimental modes (on 147.38/.98 MHz) and a machine on the new 902-MHz band,

Via **HAMNET.**

SCANNING DIGIPEATER

In a message from W6CUS-1 PBBS that was posted on CompuServe HAMNET, Don Simon, NI6A, reports that there is now a scanning digipeater serving the San Francisco area. The digipeater, N6IJP-1, is located about 2000 feet above the city of Angwin, California. It scans all packet frequencies between 145.10 MHz and 145.09 MHz, stopping for six seconds on each frequency. Once it has been accessed on a given channel, the digipeater stays on that channel until it has not been accessed for two minutes. If your group would like to have a digipeater that serves occasional contacts on several frequencies, this scheme may be the way to go. For further details on how the scanning digipeater is implemented, contact

Randy Fischer, N6IJP
455 Bay Street
Angwin, CA 94508.

Via **HAMNET.**

HF GATEWAY IN COLORADO

Dave Shavey, KØHOA, now has an HF gateway operating in Colorado Springs, Colorado. Dave is using the WØRLI Gateway software, and provides full service, including message forwarding, 24 hours a day. Like most of the other HF gateways, KØHOA is on 14.103 MHz.

We hear that the midwest is not far behind, with the crew at WB9FLW in St. Louis, Missouri, moving quickly to get a WØRLI Gateway and MailBox on the air.

Via **NØCCZ, DRNET.**

NEW CALIFORNIA DIGIPEATER

A new digipeater in northern California, **WA6YNG-1,** should provide communications between southern Oregon and Sacramento. The digipeater was installed in the Mt. Shasta area, at an elevation of 7000 feet. Both county and state Offices of Emergency Services cooperated to make this high-altitude site available. Operating on 145.01 MHz, **WA6YNG-1** can be reached through **W6AMT-7.**

Via HAMNET.

ST. LOUIS MOVES TO 145.01 MHZ

The St. Louis Amateur Packet Radio Club (SLAPR) is moving its packet operation from 147.555 MHZ to 145.01 MHz. This change has been coordinated in Eastern Missouri and Southern Illinois. Our coordination agreement states that the move will be complete by October 15th.

Via WB9FLW, DRNET.

REPRODUCTION OF GATEWAY MATERIAL

Material may be exerpted from Gateway without prior permission, provided that the original contributor is credited and Gateway is identified as the source.

Gateway: The ARRL Packet Radio Newsletter is published by the

American Radio Relay League
225 Main Street
Newington, CT 06111

Jeffrey W. Ward, K8KA
Editor

Larry E. Price, W4RA
President

David Sumner, K1ZZ
Executive Vice President

Subscriptions are available to ARRL members and nonmembers, at the following rates:

For 25 issues of Gateway:
- In the U.S.
  - ARRL Member $6 .00
  - Nonmember $9.00
- In U.S., Canada, and Mexico by First Class Mail
  - ARRL Member $11.00
  - Nonmember $14.00
- Elsewhere by Airmail
  - ARRL Member $14.00
  - Nonmember $17.00.

Prices are subject to change without notice. Remittance must be in U.S. funds, and checks must be drawn on a bank in the U.S.

Gateway: The ARRL
Packet Radio Newsletter
American Radio Relay League
225 Main St.
Newington, CT 06111

Nonprofit Organization
U.S. Postage
PAID
Hartford, Conn.
Permit No. 2929